Dayananda Sagar University

School of Engineering

Devarakaggalahalli, Harohalli, Kanakapura Road, Bangalore South Dt., Bengaluru – 562 112

Department of Computer Science & Technology

B.Tech. PROGRAMME– 2025-2029 BATCH

SCHEME AND SYLLABUS

(1th to 8th semester)

Academic Year 2026-27

Implementation of National Education Policy (NEP) 2020 for the B.Tech students of Batch 2025-2029

The implementation of Curriculum follows NEP 2020 and addresses the following features and categories of courses:

1. Student Centric flexible curriculum.
2. Inter-disciplinary Courses,
3. Multi-disciplinary Courses,
4. Ability Enhancement Courses,
5. Skill Enhancement Courses,
6. Value Added Courses,
7. Product Design and Development,
8. Internship (Rural Internship, Industry Internship, Research/Development Internship), and
9. Multiple Exit and Multiple Entry

- Certificate in Engineering after completion of first year.
- Diploma in Engineering after completion of second year.
- Advanced Diploma in Engineering after completion of third year.
- Degree in Engineering after completion of fourth year.

PART-A

GOVERNING REGULATIONS FOR BACHELOR OF TECHNOLOGY (B. Tech.) 2025-29

PREAMBLE

The School of Engineering under Dayananda Sagar University (DSU) provides Science & Technology based education leading to the development of high calibre engineers suitable for Industry and Scientific Organization. The curriculum focuses on knowledge-based course work integrated with skill development as a part of training. It equally helps in inculcating the scientific temper for the lifelong processes of learning. At the Undergraduate level, a candidate goes through the foundation courses in Science, Humanities & Engineering. Each department ensures that the courses cover both the core & electives courses, as required. Provision for Institutional elective help the candidates to acquire interdisciplinary knowledge base or specialize significantly in an area outside the parent discipline.

DEFINITIONS OF KEY WORDS

(i) **Academic Year:** Two consecutive odd, even semesters and a summer term for make up if required.

(ii) **Course:** Usually referred to as a subject, a course may consist of any of Lecture/Tutorials/Practical /Seminar/Mini project/Project work.

(iii) **Credit:** A unit by which the course work is measured. One credit is equivalent to one hour of lecture or one hour of tutorial or two hours of laboratory/practical/ workshop practice per week.

(iv) **Credit Point:** It is the product of grade point and number of credits per course.

(v) **Cumulative Grade Point Average (CGPA):** It is the measure of overall cumulative performance over all semesters. It is expressed up to two decimal places.

(vi) **First Attempt:** If a candidate has completed all formalities of academic requirement in a term and become eligible to attend the examinations and attend all the end semester examinations, such attempt shall be considered as first attempt.

(vii) **Grade Point:** It is a numerical weight allotted to each letter grade on a 10-point scale.

(viii) **Letter Grade:** It is an index of the performance in a said course. Grades are denoted by alphabets.

(ix) **Programme:** An educational activity leading to award a degree or Certificate.

(x) **Semester Grade Point Average:** It is a measure of performance during a semester. It shall be expressed up to two decimal places.

(xi) **Transcript:** Based on the grades earned, a grade certificate shall be issued after every semester to the candidate registered.

(xii) **Failure:** It is the case of appearing for Semester End Examinations but fails to obtain minimum passing marks in Semester End Examinations.

(xiii) **Detain**: It is the case of not satisfying the eligibility criteria w.r.t Attendance /Internal Assessment in each course to appear for Semester End Examination.

(xiv) **Audit Course:** A course to be taken by the student without benefit of a grade or a credit.

(xv) **Not Fit For The Program (NFFTP)**: It is the failure of satisfying the criteria laid down by regulations to continue the program of study, which leads to the termination from the University

RULES AND REGULATIONS

UG 1. All B.Tech programmes offered by the University shall be governed by the DSU B.Tech Rules and Regulations – 2023.

UG 2. The B. Tech. rules and regulations shall be applicable to any new discipline(s) that may be introduced in future.

UG 3. A candidate shall become eligible for the award of the B.Tech. Degree after fulfilling all the academic requirements as prescribed by the B.Tech. Rules and Regulations of DSU.

UG 4. ELIGIBILITY FOR ADMISSION

UG 4.1. Admission to First Year Bachelor of Technology shall be open to candidates who have passed the second year Pre-University or XII standard or equivalent examination recognized by the University.

UG 4.2. The candidate shall have studied and passed English as one of the courses and secured not less than forty five percent (45%) marks in aggregate with Physics and Mathematics as compulsory courses, along with any one of the following courses, namely, Chemistry, Bio-Technology, Computer Science, Biology and Electronics. Eligibility shall be 40% in optional courses in case of candidates belonging to SC/ST and OBC candidates from Karnataka.

UG 4.3. Admission to II year /III Semester Bachelor of Technology under Lateral entry shall be open to the candidates who have passed diploma or equivalent qualification as recognized by statutory and regulatory body.

UG 4.4. For candidates who have completed Diploma from other than State of Karnataka, their eligibility shall be based on the recognition of the Diploma awarding Boards by the University.

UG 4.5. Diploma candidates seeking admission under Lateral entry shall take up bridge courses as prescribed in the Scheme of Teaching.

UG 4.6. Admission to II year /III Semester Bachelor of Technology shall be open to candidates who have passed B. Sc. degree from a recognized University or equivalent as recognized by the University and secure not less than 45% marks in aggregate (including all semesters). Eligibility shall be 40% in case of candidates belonging to SC/ST and OBC candidates from Karnataka.

UG 4.7. B.Sc. Graduates seeking admission under Lateral entry shall take up bridge Courses as prescribed in the Scheme of Teaching.

UG 5. ACADEMIC SESSION

UG 5.1. Each academic session is divided into two semesters of approximately sixteen Weeks duration and a summer term: an odd semester (August -December), an even semester (January - May) and summer term (Make-up term) June-July.

UG 5.2. The approved schedule of academic activities for a session, inclusive of dates for registration, mid-semester and end-semester examinations, vacation breaks, shall be laid down in the Academic Calendar for the session.

UG 6. CHANGE OF BRANCH

UG 6.1. Normally a candidate admitted to a particular branch of the undergraduate programme will continue studying in that branch till completion.

UG 6.2. However, in special cases, the University may permit a candidate to change from one branch of studies to another after the first two semesters. Such changes will be permitted, in accordance with the provisions laid down hereinafter.

UG 6.3. Only those candidates will be considered eligible for change of branch after the second semester, who have completed all the credits required in the first two semesters of their studies in their first attempt, without having to pass any course requirement in the summer term examination.

UG 6.4. Applications for a change of branch must be made by intending eligible candidates in the prescribed form. The academic section will call for applications at the end of second semester of each academic year and the completed forms must be submitted by the last date specified in the notification.

UG 6.5. Candidates may enlist their choices of branch, in order of preference, to which they wish to change over. It will not be permissible to alter the choices after the application has been submitted.

UG 6.6. Change of branch shall be made strictly in the order of merit of the applicants. For this purpose, the CGPA obtained at the end of the second semester shall be considered. In case of a tie, SGPA of second semester followed by SGPA of first semester shall decide the tie.

UG 6.7. The applicants may be allowed a change in branch, strictly in order of merit, course to the limitation that the strength of a branch should not fall below the existing strength by more than ten percent and should not go above the sanctioned strength by more than ten percent. The minimum class strength of 75% should be maintained, while considering the change of branch.

UG 6.8. All changes of branch made in accordance with the above rules shall be effective from the third semester of the applicants concerned. No change of branch shall be permitted after this.

UG 7. COURSE STRUCTURE

UG 7.1. Medium of instruction, examination and project reports shall be in English except in case of any language audit courses.

UG 7.2. Teaching of the courses shall be reckoned in credits: Credits are assigned to the Courses based on the following general pattern:

(a) One credit for each lecture period.

(b) One credit for each tutorial period.

(c) One credit per two hours for each Laboratory or Practical or workshop session.

(d) Credits for seminar, mini project, project as indicated in the scheme/curriculum of teaching.

UG 7.3. In order to qualify for a B. Tech. degree of the University, a candidate is required to complete the credit requirement as prescribed in the scheme/curriculum for a particular programme.

UG 7.4. The program of a study consists of the following components:

(i) Humanities and Social Sciences including Management courses

(ii) Basic Science courses

(iii) Engineering Science courses

(iv) Professional core courses

(v) Open Electives

(vi) Project work, seminar and internship

(vii) Mandatory/Audit Courses

UG 7.5. Every B. Tech. Programme shall have a curriculum and syllabi for the courses approved by the Board of Governors. Board of Studies will discuss and recommend the syllabi of all the undergraduate courses offered by the department from time to time before sending the same to the Academic Council. Academic Council will consider the proposals from the Board of Studies and make recommendations to the Board of Management and Board of Governors for consideration and approval. For all approved courses, the copyright shall be with DSU.

UG 7.6. Faculty Advisor: To help the candidates in planning their courses of study and getting general advice on the academic programme, the concerned department will assign a Faculty Advisor to each candidate.

UG 8. REGISTRATION

UG 8.1. Every candidate is required to register for approved courses through the assigned Faculty Advisor at the end of previous semester or first week of the current semester, as notified by the Academic Calendar.

UG 8.2. The Dean may cancel the registration of one or more courses if they are found to violate some rules or if there are restrictions imposed due to disciplinary reasons.

UG 8.3. The student is permitted to drop a course/s from the registered courses, within 4 weeks after the start of the Semester/Year as notified in the academic calendar, with the permission of Faculty Advisor and Chairperson / Dean / Principal of the respective School/College and no mention will be made in the grade card for dropped courses.

UG 8.4. The student is permitted to withdraw course/s from the registered courses, within 4 weeks before the start of the Semester/Year End Examinations as notified in the academic calendar, with the permission of Faculty Advisor and Chairperson / Dean / Principal of the respective School / College and Grade "W" will be awarded for course/s that were withdrawn.

UG 8.5. For the courses with "W" grade, the students should re-register subsequently when offered, either in MOOCS or in-class or summer term and fulfil the passing criteria to secure a grade in that course for change from "W" grade.

UG 8.6. Only those candidates shall be permitted to register who have:

(a) The academic eligibility to move to higher semesters (UG 9 & UG 11)

(b) Cleared all University, Hostel and Library dues and fines (if any) of the previous semesters,

(c) Paid all required advance payments of University and Hostel dues for the current semester,

(d) Not been debarred from registering on any specific ground.

(e) A minimum CGPA of 4 in the previous semesters

UG 9. EXAMINATION: ASSESSMENT CRITERIA & ELIGIBILITYFOR PROGRESSION

Every student shall be assessed for eligibility to higher semester through Continuous Internal Assessment (CIA) and Semester End Examination (SEE) as prescribed.

UG 9.1. The Continuous Internal Assessment (CIA) shall normally be conducted by the assessment components spread through the running semester; the components of CIA may be tests, mid-term exam, quiz, term paper, simulation based problem solving, open-book test, solving open-end problems, mini- projects, seminars, viva-voce, awarding marks for attendance and such activities that enhance original thinking of students. The Course instructor shall announce the detailed methodology for conducting the various components of CIA together specifying component-wise weightages right in the commencement of each semester.

UG 9.2. The Semester End Examinations (SEE) shall be conducted at the end of each semester. The SEE components may be a closed or open book examination, project demo, viva-voce, and/or a portfolio presentation.

UG 9.3. CIA and SEE shall respectively have 60:40 percent weightage. The Vice-Chancellor, on the recommendations of the Dean of Faculty and Department Chair, in exceptional cases, may approve the variation in this weightage ratio.

UG 9.4. The performance of a student with respect to a course in a semester shall be the combined score of marks/points, he/she secures in CIA and SEE, put together. A minimum of securing 40% marks, combining both the CIA with SEE marks secured with respect to a course, shall entail the student a PASS in the course. The Vice-Chancellor, in such cases where the entire class has fared poorly in the course, upon receiving a representation by the students / department, and based on the recommendations of the committee constituted for the purpose, may review the criterion of 40%.

UG 9.5. ATTENDANCE ELIGIBILITY

UG 9.5.1. Candidates are required to attend all the classes (Lectures, Tutorials, Practical, Workshop Practice, etc.) for which they have been registered.

UG 9.5.2. The candidate shall not be allowed to appear for the end semester examination if his/her attendance falls below 85% in each course and shall be awarded a “NE” grade in that course.

UG 9.5.3. A provision for condonation of 10% of the attendance by the Vice-Chancellor on the specific recommendation of the chairman of the department and Dean, showing reasonable cause such as:

(a) Any medical emergencies/ illness where the candidate requires rest for the specified number of days certified by a Government Doctor only /any death in the family (near and dear ones).

(b) If the student represents the University in Sports/ Cultural Activities/Extra- curricular activities/Co-curricular activities.

(c) If a student presents a Paper in National/ International Conferences or attends any recognized Workshops/Seminars.

UG 9.5.4. If the period of leave is for a short duration (less than two weeks), prior application for leave shall have to be submitted to the Chairman of the Department concerned stating fully the reasons for the leave requested for along with supporting document(s). Such leave will be granted by the Chairman of the Department. However, the student shall comply with 9.5.2 and 9.5.3. of regulations.

UG 9.5.5. If the period of absence is likely to exceed two weeks, a prior application for grant of leave will have to be submitted through the Chairman of the Department to the Dean with supporting documents in each case. The decision to grant leave shall be taken by the Dean on the recommendation of the Chairman of the Department. However the student shall comply with 9.5.2 and 9.5.3.of regulations.

UG 9.5.6. It shall be the responsibility of the candidate to intimate the concerned course instructor(s) regarding his/her absence before availing the leave.

UG 9.6. CONTINUOUS INTERNAL ASSESSMENT

UG 9.6.1. Candidate shall participate in all components of Continuous Internal Assessment (CIA) to become eligible to take up the Semester End Examination or else 'NE' grade shall be awarded. However, the Vice-Chancellor, under exceptional circumstances on the recommendations of Dean of Faculty and Department Chair, may exempt a student from participation in CIA component/s and permit taking up SEE.

UG 9.6.2. There shall be no marks improvement of Continuous Internal Assessment; however, the withdrawal and re-registering of the course shall be permitted.

UG 9.6.3. Continuous Evaluation consists of:

UG 9.6.3.1. Under normal circumstances for theory courses, total CIA weightage shall be a total of 60%, put together all components with varying weightages; Under exceptional circumstances with the approval of the Vice-Chancellor on the recommendation of Dean of the School, the weightage of CIA may be lower/higher than 60%.

The components of CIA may be tests, mid-term exam, quiz, term paper, simulation based problem solving, open-book test, solving open-end problems, mini-projects, seminars, viva-voce, awarding marks for attendance and such activities that enhance original thinking of students.

UG 9.6.3.2 Under normal circumstances for the practical courses (laboratory, workshops, and any such hands-on activity), total CIA weightage shall be a total of 60%, put together all components with varying weightages; Under exceptional circumstances with the approval of the Vice-Chancellor on the recommendation of Dean of the School, the weightage of CIA may be lower/higher than 60%.

CIA may have components such as conduction of an experiment, record writing, viva-voce, tests, simulation, mid-term exam, quiz, demo, term paper, mini-projects, seminars, marks for attendance and activities which enhances original thinking of students.

UG 10. GRADING

UG 10.1 There shall be continuous assessment of a candidate's performance throughout the semester and grades shall be awarded by the concerned course instructor and/or the appropriate committee appointed for this purpose on the following basis.

UG 10.2 The grading will normally be based on CIA and SEE.

UG 10.3. Practical Courses/ Workshop Practice: The evaluation will be based on instructor's continuous internal assessment, a test and end semester examination.

UG 10.4. The weightage assigned to different components of continuous internal assessment will be announced by the concerned instructor(s) in the beginning of the semester.

UG 10.5. The results of performance of the candidates in the Continuous Internal assessment Test shall be announced by the instructors.

UG 10.6. In case of seminar, evaluation will be as determined by the grade awarding Committee (as per the Program scheme).

UG 10.7. Mini project /projects will be based on Continuous evaluation by Guide(s) and Semester End Examination (as per the Program scheme)

UG 10.8. The results of performance of the candidates shall be announced by the Controller of Examinations.

UG 10.9. METHOD OF AWARDING LETTER GRADES

UG 10.9.1. Relationships among Grades, Grade points and % of marks are listed in Table1.

UG 10.10. DESCRIPTION OF GRADES

UG 10.10.1 Table 1 shows the relationships among the grades, grade points and percentage of marks.

UG 10.10.2. A student will have to ensure a minimum CGPA of 4, to become eligible for the award of the degree.

UG 10.10.3. A candidate shall have to repeat all courses in which he/she obtains 'F' Grades until a passing grade is obtained.

Table 1: Grade, Points, Grade Description and % of marks

GRADE	GRADE POINTS	DESCRIPTION	% MARKS
O	10	Outstanding	90 to 100
A+	9	Excellent	80 to 89
A	8	Very Good	70 to 79
B+	7	Good	60 to 69
B	6	Above Average	55 to 59
C	5	Average	50 to 54
P	4	Pass	40 to 49
F	0	Fail	< 40
AP	-	Audit Pass	-
AF	-	Audit Fail	-
IC	-	In Complete	-
NE	-	Not Eligible	-
W	-	Withdrawn	-

UG 10.10.4. An IC grade denotes incomplete performance in any Theory and/or Practical Assessment. It may be awarded in case of absence on medical grounds or other special circumstances for SEE. Requests for IC grade should be made at the earliest but not later than the last day of SEE.

UG 10.10.5. The student can appear for the course/s with IC grade, when exams are conducted subsequently by the University for those Courses.

UG10.11. EVALUATION OF PERFORMANCE

UG10.11.1. The performance of a candidate shall be evaluated in terms of the Semester Grade Point Average (SGPA) which is the Grade Point Average for a semester, Cumulative Grade Point Average (CGPA) which is the Grade Point Average for all the completed semesters.

UG 10.11.2. The Earned Credits (EC) are defined as the sum of course credits for courses in which candidates have been awarded grades between O to P. (Table 1)

UG 10.11.3. Points earned in a semester = (Course credits X Grade point) for Grades O - P

UG 10.11.4. The SGPA is calculated on the basis of grades obtained in all courses, except audit courses and courses in which F grade or below, registered for in the particular semester.

$$\text{SGPA} = \frac{\text{Points secured in the semester (O – P Grades)}}{\text{Credits registered in the semester, excluding audit}}$$

UG 10.11.5. The CGPA is calculated on the basis of all pass grades, except audit courses.

$$\text{CGPA} = \frac{\text{Cumulative points secured in all the passed courses (O – P Grades)}}{\text{Cumulative registered credits, excluding audit}}$$

UG 10.12. WITHHOLDING OF GRADES

UG 10.12.1. Grades shall be withheld when the candidate has not paid his/her dues or when there is a disciplinary action pending against him/her.

UG 10.13. CONVERSION OF CGPA INTO PERCENTAGE

UG 10.13.1. Conversion formula for the conversion of CGPA into percentage is Percentage of Marks Scored = (CGPA Earned − 0.75) × 10

UG 11. PROMOTION CRITERIA AND ENROLLMENTS TO HIGHER SEMESTERS

UG 11.1. During registration to the higher semesters, the following criteria/conditions for promotion, shall be satisfied.

UG 11.1.1. A student shall 'Not Eligible' (NE) for writing SEE if he/she does not comply to the minimum prescribed attendance in any course that carry a credit.

Students shall register afresh for such course/s, whenever offered next, to meet the attendance requirements and secure a pass grade, subsequently in that course/s.

UG 11.1.2. In a semester (ODD / EVEN), a student is deemed to be Not Eligible (NE) if he/she does not satisfy minimum attendance requirements criteria in a credit course.

If this course happens to be a prerequisite to a connected course in the subsequent semester, then the student shall not be permitted to register for that connected course until he / she secures pass grade in the prerequisite course by complying to the minimum attendance requirement when the prerequisite course is offered next (either during summer term or regular semester).

UG 11.1.3. A student shall be permitted to register for FOUR credited courses or to a total of 16 credits whichever is higher along with pending audit courses, if any, during a summer term by paying the prescribed course registration fee per credit notified by the university from time to time.

UG 11.1.4. The students with NE ('NOT ELIGIBLE' due to shortage in attendance) in any Credit Course/s other than Audit Courses in a semester shall have to secure a pass grade by compliance to minimum attendance requirements in the NE course to register for connected courses if NE course happens to be prerequisite course for those connected courses offered in the subsequent semesters.

UG 11.1.5. Candidates who secure 'F' grade in any courses in regular semester or summer term shall secure PASS grade in such course/s either in the subsequent summer term examination or shall repeat in the next appropriate semester whenever it is/they are offered, i.e. odd semester courses during odd semesters examinations and even semester courses during even semester examinations, respectively.

UG 11.2. In case of failure in Practical/Workshop practice course the candidate in any semester may clear it in the subsequent summer term examination or semester examination.

UG 11.3. In case a candidate fails in Practical/ Workshop practice he/she shall register when it is offered next either in the summer term or subsequent semester, as the case may be.

UG 11.4. Candidates may add and drop course(s) with the concurrence of the Faculty Advisor, and under intimation to the concerned course instructors and the academic section provided this is done within the date mentioned in the Academic Calendar.

UG 11.5. SUMMER TERM & MAKEUP EXAMINATIONS

UG 11.5.1. A summer term program may be offered by a department and with the approval of the Dean.

UG 11.5.2. Summer term courses will be announced by the Academic Affairs Office at the end of the even semester and before the commencement of the end semester examination. A candidate will have to register for summer term courses by paying the prescribed fees within the stipulated time in the announcement.

UG 11.5.3. The total number of contact hours in any summer term program will be the same as in the regular semester course. The assessment procedure in a summer term course will also be like the procedure for a regular semester course.

UG 11.5.4. Candidates granted semester drop by the Board of Governors, on medical ground, shall be allowed to clear the concerned courses in summer term course and subject to conditions as stated under clauses 11.5.1, 11.5.2.and 11.5.3.

UG 11.5.5. The Candidates with "NE" grade shall register for summer term by paying the prescribed fees.

UG 11.5.6. Candidates who are awarded 'F' grades in regular semester examinations have the option to register for the concerned courses in summer term examinations to the conditions as stated under clauses 11.5.1, 11.5.2.and 11.5.3above, or they can re-sit for subsequent semester/summer term examination only.

UG 11.5.7. Provision for make-up exam shall be available to the students who might have missed to attend the Semester / Annual end examinations of one or more courses for exceptional cases arising out of natural calamities / medical emergencies / death of a member in the family, with the permission of Faculty Advisor and Chairperson / Dean / Principal of the respective School/College. All such cases have to be exclusively to be approved by the Vice-Chancellor and ratified in the Academic Council / BOM / BOG. All such courses approved for makeup examinations are awarded a transitory grade "IC" (incomplete grade)

UG 11.5.8. The makeup examinations shall be held as notified in the academic calendar or through an exclusive notification duly approved by the Vice-chancellor.

UG 12. DURATION OF THE PROGRAMME

UG 12.1. Normally a candidate should complete all the requirements for undergraduate programme in four years. However, academically weaker candidates who do not fulfil some of the requirements in their first attempt and have to repeat them in subsequent semesters may be permitted up to eight consecutive years (from the first year of registration) to complete all the requirements of the degree.

UG 12.2. Normally a candidate under lateral entry should complete all the requirements for undergraduate programme in three years. However, academically weaker candidates who do not fulfil some of the requirements in their first attempt and have to repeat them in subsequent semesters may be permitted up to six consecutive years (from the second-year registration) to complete all the requirements of the degree.

UG 13. TERMINATION FROM THE PROGRAMME

UG 13.1. A candidate may also be compelled to leave the Program in the University on disciplinary grounds.

UG 13.2. On having been found to have produced false documents or having made false declaration at the time of seeking admission.

UG 13.3. On having been found to be pursuing regular studies and/or correspondence courses (leading to degree or diploma) in any other college, university, or an educational institution simultaneously.

UG 13.4. On having been found to be concurrently employed and performing duty or carrying out business in contravention to academic schedules of the University and without seeking approval from the University.

UG 13.5. If a student fails to earn a pass grade even after 4 attempts such a student is terminated from the university on the grounds of NOT FIT FOR THE PROGRAM (NFFTP).

UG 13.6. If a student secures a CGPA less than 4.0, 4 times during entire duration of the program of study, such a student is terminated from the university on the grounds of NOT FIT FOR THE PROGRAM (NFFTP).

UG 13.7. However, if the student appeals for reconsideration of termination from the university under NFFTP rule by providing the guanine reasons to the Vice- Chancellor through the Dean of Faculty, then the Vice-Chancellor may consider constituting a committee for the purpose of review and provide 2 additional attempts on the recommendations of the committee.

UG 14. TEMPORARY WITHDRAWAL FROM THE UNIVERSITY

UG 14.1. Candidate who has been admitted to an undergraduate programme of the University may be permitted to withdraw temporarily from the University on the grounds of prolonged illness or grave calamity in the family for a period of one semester or more, provided:

UG 14.1.1. He/she applies to the University within at least 6 weeks of the commencement of the semester or from the date he last attended his/her classes whichever is later, stating fully the reasons for such withdrawal together with supporting documents and endorsement of his/her guardian.

UG 14.1.2. The University is satisfied that, counting the period of withdrawal, the candidate is likely to complete his/her requirements of the B. Tech. Degree within the time limits specified in Clause 12.1 or 12.2 above.

UG 14.1.3. There are no outstanding dues or demands in the University / Hostel / Department / Library.

UG 14.1.4. Normally, a candidate will be permitted only one such temporary withdrawal during his/her tenure as a candidate of the undergraduate programme.

UG 15. TRANSFER OF CANDIDATES

UG 15.1. Transfer of candidates from higher education institutions outside university shall be considered at the beginning of Third and Fifth Semesters but subject to confirmation of equivalence.

UG 15.2. The candidates shall apply for equivalence with the No-objection for admission to DSU from the University where they are perusing their study.

UG 15.3. The candidates must have passed in all courses in the earlier semesters prior to transfer.

UG 16. ELIGIBILITY FOR THE AWARD OF B. Tech. DEGREE

A candidate shall be declared to be eligible for the award of B. Tech. degree if he/she has:

UG 16.1. Completed all the credit requirements for the degree with a CGPA 4.0 or higher at the end of the programme.

UG 16.2. Satisfactorily completed all the mandatory audit courses.

UG 16.3. No dues to the University, Department, Hostels.

UG 16.4. No disciplinary action pending against him/her.

UG 17. AWARD OF DEGREE

The award of B. Tech. degree must be recommended by the Academic Council and approved by the Board of Management and Board of Governors of the DSU.

UG 18. CONDUCT AND DISCIPLINE

UG 18.1. Candidates shall conduct themselves within and outside the precincts of the University in a manner befitting the candidates of an institution of national importance. The University has a separate ordinance Code and Conduct of Candidates which is applicable to all candidates of the University.

UG 19. REPEAL AND SAVINGS

Notwithstanding anything contained in these Regulations, the provisions of any guidelines, orders, rules or regulations in force at the University shall be inapplicable to the extent of their inconsistency with these Regulations. The Academic Council, Board of Management and Board of Governors of Dayananda Sagar University may revise, amend or change the regulations from time to time.

UG 20. INTERPRETATION

Any questions as to the interpretation of these Regulations shall be decided by the University, whose decision shall be final. The University shall have the powers to issue clarifications to remove any doubt, difficulty or anomaly which may arise during the implementation of the provisions of these regulations.

General Instructions:

- **Open Elective Courses:** At least two courses must be provided from each department and the courses shall be general course on emerging areas with broad coverage of syllabus so that student shall chose without any difficulty.
- **Honors Degree:** An Honors degree typically refers to a higher level of academic achievement in the major area. That is, certificate in his/her OWN major for Research orientation. The Credit requirement: **178 credits** (Major worth 160 credits + Honors 18 credits)
- **Minor Degree:** Minor is a secondary concentration of courses that often complements the honors. Minor in any OTHER branch for Improving Employability.
  - Minor is an option rather than a requirement for B. Tech students. They may opt for one of the Engineering or Non-engineering discipline as Minor, earning additional credits ranging from 18. However, students are permitted to choose only one Minor either from engineering or non-engineering discipline.
  - This opportunity is ideal for students who took a Major out of necessity but would still like to pursue their passion in another discipline or to enrich/equip them for a specific profession where greater job opportunities exist. Another advantage of opting for a Major with a Minor is to earn standing credits for pursuing a Master’s degree abroad or within India too.
  - Only students who satisfy a set of minimum eligibility criteria set forth by the university and meet certain pre-requisites, will be permitted to opt for a Minor.
  - Credit requirement: **178 credits** (Major worth 160 credits + Minor 18 credits)
  - Degree nomenclature: The degree will contain the Major / Major with Specialization. The Minor pursued by the student will be provided in the transcript along with details on courses completed and associated credits earned.
  - For e.g., For a student who pursued Computer Science and Engineering with a Minor in Industrial Psychology, the degree will read “B. Tech in Computer Science and Engineering”, Transcripts of B. Tech will reflect the Minor courses and the Minor certificate in Industrial Psychology will be issued separately.

PART-B

SCHEME AND SYLLABUS FOR B.TECH. CST (FIRST TO FOURTH YEARS) PROGRAMME

Definitions / Descriptions

Definition of Credit:	
1 Hour Lecture (L) Per Week	01 Credit
1 Hour Tutorial (T) Per Week	0.5 Credit
1 Hour Practical (P) Per Week	0.5 Credit
1 Hour Project (J) Per Week	0.5 Credit

Course code and Definition:	
BSC	Basic Science Courses
ESC	Engineering Science Courses
HSMC	Humanities and Social Sciences including Management Courses
IPCC	Integrated Professional Core Course
PCC	Professional Core Courses
PEC	Professional Elective Courses
OEC	Open Elective Courses
SEC	Skill Enhancement Courses
UHV	Universal Human Value Course
PROJ	Project Work
INT	Internship

SCHEME 2025 – 2029 Batch

I SEMESTER (CHEMISTRY CYCLE)

S.NO	Course Type	Course Code	Course Name	Teaching Hours / Week				Examination			Credits
				Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
				L	T	P	J				
1	BSC	25EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	3	1	0	0	60	40	100	**4**
2	ESC	25EN1102	ENGINEERING GRAPHICS AND DESIGN THINKING	2	1	2	0	60	40	100	**4**
3	BSC	25EN1103	CHEMISTRY FORCOMPUTER SCIENCE CLUSTER	2	1	2	0	60	40	100	**4**
4	BSC	25EN1104	CHEMISTRY FOR ECE	2	1	2	0	60	40	100	**4**
5	BSC	25EN1105	CHEMISTRY FOR MECHANICAL CLUSTER	2	1	2	0	60	40	100	**4**
6	BSC	25EN1106	CHEMISTRY FOR CSME	2	1	2	0	60	40	100	**4**
7	ESC	25EN1107	ENGINEERING SCIENCE COURSE –I for Computer science Cluster Object Oriented Programming	3	0	0	0	60	40	100	**3**
8	ESC	25EN1108	ENGINEERING SCIENCE COURSE– I for ECE Introduction to Electrical Engineering	3	0	0	0	60	40	100	**3**
9	SEC	25EN1109	ENGINEERING SCIENCE COURSE – I for Mechanical Cluster Introduction to Electrical and Electronics Engineering	3	0	0	0	60	40	100	**3**
10	BSC	25EN1110	TECHNICAL ENGLISH	2	0	0	0	100	-	100	**2**
11	HSMC	25EN1111	ENVIRONMENTAL SCIENCE	2	0	0	0	100	-	100	**2**
12	HSMC	25EN1112	KANNADA KALI / MANASU	1	0	0	0	50	-	50	**1**
13	SEC	25EN1113	COGNITIVE AND TECHNICAL SKILLS -I								**P/F**
			Total	**15**	**3**	**4**	**0**	**490**	**160**	**650**	**20**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

ENGINEERING SCIENCE COURSE – I: Computer Science Cluster: Object Oriented Programming

ECE: Introduction to Electrical Engineering **Mechanical Cluster:** Introduction to Electrical and Electronics Engineering

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South District, Karnataka – 562 112

I SEMESTER (PHYSICS CYCLE)

S.NO	Course Type	Course Code	Course Name		Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project		CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	BSC	25EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS		3	1	0	0		60	40	100	4
2	ESC	25EN1114	C PROGRAMMING FOR PROBLEM SOLVING		2	1	2	0		60	40	100	4
3	BSC	25EN1115	PHYSICS FOR COMPUTER SCIENCE CLUSTER		3	0	2	0		60	40	100	4
4	BSC	25EN1116	PHYSICS FOR ECE		3	0	2	0		60	40	100	4
5	BSC	25EN1117	PHYSICS FOR MECANICAL CLUSTER		3	0	2	0		60	40	100	4
6	ESC	25EN1109	ENGINEERING SCIENCE COURSE – II for Computer Science cluster Introduction to Electrical and Electronics Engineering		3	0	0	0		60	40	100	3
7	ESC	25EN1118	ENGINEERING SCIENCE COURSE – II for ECE Introduction to Electronics Engineering		3	0	0	0		60	40	100	3
8	ESC	25EN1119	ENGINEERING SCIENCE COURSE – II for Mechanical cluster Engineering Mechanics		3	0	0	0		60	40	100	3
9	ESC	25EN1120	INTRODUCTION TO SUSTAINABLE ENGINEERING		2	0	0	0		100	-	100	2
10	ESC	25EN1121	BIOLOGY FOR ENGINEERS		2	0	0	0		100	-	100	2
11	UHV	25EN1122	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS		1	0	0	0		50	-	50	1
12	SEC	25EN1113	COGNITIVE AND TECHNICAL SKILLS – I								-		P/F
			Total		**15**	**2**	**4**	**0**		**490**	**160**	**650**	**20**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

ENGINEERING SCIENCE COURSE – II:

Compute Science Cluster: Introduction to Electrical and Electronics Engineering

ECE: Introduction to Electronics Engineering

Mechanical Cluster: Engineering Mechanics

II SEMESTER (CHEMISTRY CYCLE)

S.NO	Course Type	Course Code	Course Name	Teaching Hours / Week: Lecture (L)	Tutorial (T)	Practical (P)	Project (J)	Examination: CIE Marks	SEE Marks	Total Marks	Credits
1	BSC	25EN1201	SINGLE AND MULTIVARIATE CALCULUS	3	1	0	0	60	40	100	**4**
2	ESC	25EN1102	ENGINEERING GRAPHICS AND DESIGN THINKING	2	1	2	0	60	40	100	**4**
3	BSC	25EN1103	CHEMISTRY FORCOMPUTER SCIENCE CLUSTER	2	1	2	0	60	40	100	**4**
4	BSC	25EN1104	CHEMISTRY FOR ECE	2	1	2	0	60	40	100	**4**
5	BSC	25EN1105	CHEMISTRY FOR MECHANICAL CLUSTER	2	1	2	0	60	40	100	**4**
6	BSC	25EN1106	CHEMISTRY FOR CSME	2	1	2	0	60	40	100	**4**
7	ESC	25EN1107	ENGINEERING SCIENCE COURSE– I for Computer science Cluster Object Oriented Programming	3	0	0	0	60	40	100	**3**
8	ESC	25EN1108	ENGINEERING SCIENCE COURSE – I for ECE Introduction to Electrical Engineering	3	0	0	0	60	40	100	**3**
9	ESC	25EN1109	ENGINEERING SCIENCE COURSE – I for Mechanical Cluster Introduction to Electrical and Electronics Engineering	3	0	0	0	60	40	100	**3**
10	BSC	25EN1110	TECHNICAL ENGLISH	2	0	0	0	100	–	100	**2**
11	AEC	25EN1111	ENVIRONMENTAL SCIENCE	2	0	0	0	100	–	100	**2**
12	BSC	25EN1112	KANNADA KALI / MANASU	1	0	0	0	50	–	50	**1**
			Total	**15**	**3**	**4**	**0**	**490**	**160**	**650**	**20**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

ENGINEERING SCIENCE COURSE – I:

Computer Science Cluster: Object Oriented Programming

ECE: Introduction to Electrical Engineering

Mechanical Cluster: Introduction to Electrical and Electronics Engineering

II SEMESTER (PHYSICS CYCLE)

S.NO	Course Type	Course Code	Course Name		Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project		CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	BSC	25EN1201	SINGLE AND MULTIVARIATE CALCULUS		3	1	0	0		60	40	100	4
2	ESC	25EN1114	C PROGRAMMING FOR PROBLEM SOLVING		2	1	2	0		60	40	100	4
3	BSC	25EN1115	PHYSICS FOR COMPUTER SCIENCE CLUSTER		3	0	2	0		60	40	100	4
4	BSC	25EN1116	PHYSICS FOR ECE		3	0	2	0		60	40	100	4
5	BSC	25EN1117	PHYSICS FOR MECANICAL CLUSTER		3	0	2	0		60	40	100	4
6	ESC	25EN1109	ENGINEERING SCIENCE COURSE – II for Computer Science cluster Introduction to Electrical and Electronics Engineering		3	0	0	0		60	40	100	3
7	ESC	25EN1118	ENGINEERING SCIENCE COURSE – II for ECE Introduction to Electronics Engineering		3	0	0	0		60	40	100	3
8	ESC	25EN1119	ENGINEERING SCIENCE COURSE – II for Mechanical cluster Engineering Mechanics		3	0	0	0		60	40	100	3
9	ESC	25EN1120	INTRODUCTION TO SUSTAINABLE ENGINEERING		2	0	0	0		100	-	100	2
10	AEC	25EN1121	BIOLOGY FOR ENGINEERS		2	0	0	0		100	-	100	2
11	HS MC	25EN1122	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS		1	0	0	0		50	-	50	1
12	SEC	25EN1202	COGNITIVE AND TECHNICAL SKILLS –II								-		P/F
			Total		**15**	**2**	**4**	**0**		**490**	**160**	**650**	**20**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

ENGINEERING SCIENCE COURSE – II:

Compute Science Cluster: Introduction to Electrical and Electronics Engineering

ECE: Introduction to Electronics Engineering

Mechanical Cluster: Engineering Mechanics

III SEMESTER												
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination			Credits
					Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J				
1	BSC	25CT2301	Applied Statistics & Stochastic Processes	MAT	3	0	0	0	60	40	100	**3**
2	PCC	25CT2302	Logic Design & Computer Organization Architecture	CST	3	0	0	0	60	40	100	**3**
3	IPCC	25CT2302	Database Management System	CST	3	0	2	0	60	40	100	**4**
4	IPCC	25CT2304	Data Structures Using C	CST	2	0	2	0	60	40	100	**3**
5	IPCC	25CT2305	Sensors, Systems & IoT Fundamentals	CST	3	0	2	0	60	40	100	**4**
6	PCC	25CT2306	Sustainability Engineering & ESG	CST	2	0	0	0	60	40	100	**2**
7	SEC	25CT23XX	Skill Enhancement Course I	CST	1	0	0	2	100	--	100	**2**
8	SEC	25CT2311	Cognitive and Technical Skills – III	All Dept.	0	0	0	0	--	--	--	**P/ F**
			Total		**17**	**0**	**6**	**2**	**460**	**240**	**700**	**21**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

Skill Enhancement Course – I						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT2307	AI Powered -Full Stack Web Development	1	0	0	2
2	25CT2308	Git Hub & Version Control	1	0	0	2
3	25CT2309	Indian Knowledge Systems	2	0	0	0
4	25CT2310	Low Code/No Code Development	1	0	0	2
5	XXXXXX	MOOC	2	0	0	0

IV SEMESTER													
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks		
					L	T	P	J					
1	BSC	25CT2401	Differential Equations & Optimization	MAT	3	0	0	0	60	40	100		**3**
2	PCC	25CT2402	Operating Systems for GPUs	CST	3	0	0	0	60	40	100		**3**
3	IPCC	25CT2403	Design and Analysis of Algorithms	CST	3	0	2	0	60	40	100		**4**
4	IPCC	25CT2404	Machine Learning	CST	3	0	2	0	60	40	100		**4**
5	SEC	25CT2405	Design Studio II: Innovation Design Thinking & IP	CST	1	0	0	2	60	40	100		**2**
6	PCC	25CT2406	Introduction to Data Science	CST	2	0	0	0	60	40	100		**2**
7	HSMC	25LS00XX	Humanities & Social Sciences: Liberal Studies	All dept.	2	0	0	0	100	--	100		**2**
8	HSMC	25CT24XX	Skill Enhancement Course II	CST	1	0	0	2	100	--	100		**2**
9	SEC	25CT2412	Cognitive and Technical Skills – IV	All Dept.	0	0	0	0	--	--	--		**P/F**
Total					**18**	**0**	**4**	**4**	**560**	**240**	**800**		**22**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

Skill Enhancement Course – II						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT2407	OOP Using Java	1	0	0	2
2	25CT2408	DevOps Fundamentals & Product Launch	1	0	0	2
3	25CT2409	R Programming	1	0	0	2
4	25CT2410	Introduction to Robotics	1	0	0	2
5	25CT2411	Accelerated Computing with CUDA	1	0	0	2

Regular Liberal Studies Courses

Sl. No.	Course Code	Course Title
1	25LS0001	Introduction to Japanese Language
2	25LS0002	Introduction to German Language
3	25LS0003	Law for Engineers
4	25LS0004	Introduction to Painting
5	25LS0005	Vedic Mathematics
6	25LS0006	Fundamentals of Critical Thinking
7	25LS0007	Cyber Crimes, Policies & Laws
8	25LS0008	Introduction to Film Studies
9	25LS0009	Practicing Yoga & Meditation
10	25LS0010	Universal Human Values
11	25LS0011	Data Governance

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Enhancing Soft Skills and Personality [NPTEL]
2	*CID	Soft Skill Development [NPTEL]

Note: CID = NPTEL Course ID

V SEMESTER												
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination			Credits
					Lecture L	Tutorial T	Practical P	Project J	CIE Marks	SEE Marks	Total Marks	
1	PCC	25CT3501	Responsible AI, Ethics & Policy	CST	2	0	0	0	60	40	100	**2**
2	IPCC	25CT3502	Deep Learning	CST	3	0	2	0	60	40	100	**4**
3	IPCC	25CT3503	Computer Networks & Distributed Systems	CST	2	1	2	0	60	40	100	**4**
4	PCC	25CT3504	AI-Enabled Software Systems and Agile Project Management	CST	3	0	0	0	60	40	100	**3**
5	PCC	25CT3505	Small E-Business Launch	CST	1	0	0	2	60	40	100	**2**
6	SEC	25CT3506	Design Studio II: AI-Enabled Systems, Digital–Physical Systems	CST	1	0	0	2	100	--	100	**2**
7	PEC	25CT35XX	Professional Elective I / MOOC	CST	3	0	0	0	60	40	100	**3**
8	SEC	25CT3512	Cognitive and Technical Skills – V	Any Dept.	0	0	0	0	--	--	--	**P/F**
Total					**15**	**1**	**4**	**4**	**460**	**240**	**700**	**20**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

NOTE: For Professional Elective Courses (PEC), the LTPJ division is given on the next page according to the type of course.

Professional Elective Course – I / MOOC/ Certification						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT3507	Data Science & Visualization	2	0	0	2
2	25CT3508	Cyber Security Fundamentals	2	0	0	2
3	25CT3509	Interactive Art and Creative Coding	2	0	0	2
4	25CT3510	Introduction to Quantum Computing	3	0	0	0
5	25CT3511	Product Strategy & Management	3	0	0	0

MOOC Courses

Sl. No.	**Course Code**	**Course Title**
1	*CID	Data Analytics and Visualization with MS PowerBI [IIMB] 6 WEEKS Data Analytics with Python [NPTEL] 12 WEEKS
2	*CID	Practical Cyber Security for Cyber Security Practitioners, IIT Kanpur [NPTEL] 12 WEEKS
3	*CID	Quantum Computation and Quantum Information [NPTEL] 12 WEEKS
4	*CID	Product and Brand Management, IIT Roorkee [NPTEL] 12 WEEKS

Note: CID = NPTEL Course ID

VI SEMESTER													
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination			Credits
					Lecture L	Tutorial T	Practical P	Project J	CIE Marks	SEE Marks	Total Marks	
1	PCC	25CT3601	Sustainable Software Engineering	CST	3	0	0	0	60	40	100	**3**
2	PCC	25CT3602	Lean Start-up Methodology and Product Marketing	CST	3	0	0	2	60	40	100	**4**
3	IPCC	25CT3603	Natural Language Processing for GenAI	CST	2	0	2	0	60	40	100	**3**
4	PCC	25CT3604	Theory of Computation & Compiler Design	CST	2	1	0	0	60	40	100	**3**
5	PEC	25CT36XX	Professional Elective II / MOOC	CST	3	0	0	0	60	40	100	**3**
6	PROJ	25CT3610	Minor Project	CST	0	0	0	4	100	--	100	**2**
7	OEC	25OE36XX	Open Elective I	All. Depts	3	0	0	0	60	40	100	**3**
8	SEC	25CT3611	Cognitive And Technical Skills – VI	Any Dept.	0	0	0	0	--	--	--	**P/F**
			Total		**16**	**1**	**2**	**6**	**460**	**240**	**700**	**21**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

NOTE1: For Professional Elective Courses (PEC), the LTPJ division is given on the next page according to the type of course.

NOTE2: Internship: All the students admitted to III year shall have to undergo mandatory internship of 4 weeks during the VII and VIII semesters. A university examination shall be conducted during VIII semester and the prescribed credit shall be included in VIII semester. Internship shall be considered as a head of passing and shall be considered for the award of degree. Those, who do not take up/complete the internship shall be declared fail and shall have to complete during subsequent University examination after satisfying the internship requirements.

Open Electives-I						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25OE36XX	Small E-Business Launch	2	0	0	2
2	25OE36XX	Lean Startup Methodology	2	0	0	2

Professional Elective Course – II / MOOC/ Certification						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT3605	Computer Vision	3	0	0	0
2	25CT3606	Cryptography & Cyber Security	2	0	0	2
3	25CT3607	Gaming Design Fundamentals	2	0	0	2
4	25CT3608	Quantum Algorithms	2	0	0	2
5	25CT3609	Business Management using Cloud	3	0	0	0

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Modern Computer Vision, IIT Madras [NPTEL] 12 WEEKS
2	*CID	Cryptography and Network Security, IIT Kharagpur [NPTEL] 12 WEEKS
3	*CID	Quantum Computing: Algorithms and Limitations Through the Query Model [NPTEL] 12 WEEKS

Note: CID = NPTEL Course ID

VII SEMESTER												
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination			Credits
					Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J				
1	PCC	25CT4701	Entrepreneurial Finance & Deep-Tech Start-ups	CST	2	0	0	0	100	--	100	2
2	PCC	25CT4702	AI in Healthcare	CST	3	0	0	2	60	40	100	4
3	PEC	25CT47XX	Professional Elective III	CST	3	0	0	0	60	40	100	3
4	PEC	25CT47XX	Professional Elective IV	CST	3	0	0	0	60	40	100	3
5	OEC	25OE47XX	Open Elective II	All. Dept.	3	0	0	0	60	40	100	3
6	PROJ	25CT4713	Capstone Project I	CST	0	0	0	8	60	40	100	4
			Total		**14**	**0**	**0**	**10**	**400**	**200**	**600**	**19**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

NOTE: For Professional Elective Courses (PEC), the LTPJ division is given on the next page according to the type of course.

Open Electives-II						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25OE47XX	Innovation for Sustainable Business Management	2	0	0	2
2	25OE47XX	Product Analytics	2	0	0	2

Professional Elective Course – III / MOOC/ Certification						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT4703	Building LLM with Prompt engineering	3	0	0	0
2	25CT4704	IoT-Based Data Analytics for Health Care	2	0	0	2
3	25CT4705	Game Development using Unity/ Game Development Using Godot	2	0	0	2
4	25CT4706	Quantum Machine Learning	2	0	0	2
5	25CT4707	Business to Business Marketing	3	0	0	0

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Introduction To Internet Of Things, IIT Kharagpur [NPTEL]

Note: CID = NPTEL Course ID

Professional Elective Course – IV / MOOC/ Certification						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT4708	Generative AI with Diffusion Models	3	0	0	0
2	25CT4709	Blockchain Technology	2	0	0	2
3	25CT4710	AR/VR & Game-Theoretic/ Algorithmic Game Theory	2	0	0	2
4	25CT4711	Quantum Artificial Intelligence	2	0	0	2
5	25CT4712	Sustainable Innovation Business	3	0	0	0

MOOC Courses

Sl. No.	**Course Code**	**Course Title**
1	*CID	Blockchain and its Applications, IIT Kharagpur [NPTEL] 12 WEEKS

Note: CID = NPTEL Course ID

VIII SEMESTER												
S.NO	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination			Credits
					Lecture	Tutorial	Practical	Project	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J				
1	PEC	25CT4801	Leadership, Policy & Authentic Intelligence	CST	2	0	0	0	100	--	100	**2**
2	PCC	25CT48XX	Professional Elective V / MOOC	CST	3	0	0	0	60	40	100	**3**
3	INT	25CT4807	Internship	CST	0	0	0	4	100	--	100	**2**
4	PROJ	25CT4808	Capstone Project II	CST	0	0	0	20	60	40	100	**10**
			Total		**5**	**0**	**0**	**24**	**320**	**80**	**400**	**17**

L – Lecture, T – Tutorial, P – Practical, J – Project, CIE – Continuous Internal Evaluation, SEE- Semester End Examinations.

NOTE 1: Internship

Completed during the intervening vacations of VI and VII semesters and /or VII and VIII semesters or VIII semesters

Professional Elective Course – V / MOOC/ Certification						
S.NO	**Course Code**	**Course Name**	**L**	**T**	**P**	**J**
1	25CT4802	Agentic AI and RAG Engineering	3	0	0	0
2	25CT4803	Business Intelligence & Analytics	2	0	0	2
3	25CT4804	Unreal Engine Game Development for Beginners	2	0	0	2
4	25CT4805	Quantum Cryptography	2	0	0	2
5	25CT4806	Digital Business	3	0	0	0

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Quantum Algorithms and Cryptography, IIT Madras [NPTEL] 12 WEEKS
2	*CID	Foundation of Digital Business, IIT Kharagpur [NPTEL] 8 WEEKS Business Intelligence & Analytics[NPTEL] 4 WEEKS

Note: CID = NPTEL Course ID

NOTE: Total Credits (I-Sem to VIII Sem) = 160 credits.

Year	Sem	Nu. of Credits	Per Year Credits	Total Credits
I	Sem 1	20	40	160
	Sem 2	20		
II	Sem 3	21	43	
	Sem 4	22		
III	Sem 5	20	41	
	Sem 6	21		
IV	Sem 7	19	36	
	Sem 8	17		

Elective courses domain-wise

S.N	Domain-wise	Domain Clusters	PROFESSIONAL ELECTIVE COURSES				
			PEC-I	PEC-II	PEC-III	PEC-IV	PEC-V
			5^{th} Semester	6^{th} Semester	7^{th} Semester		8^{th} Semester
			Course Name	Course Name	Course Name	Course Name	Course Name
1	Domain-1	Cognitive AI & Intelligent Systems	Data Science & Visualisation	Computer Vision	Building LLM with Prompt engineering	Generative AI with Diffusion Models	Agentic AI and RAG Engineering
2	Domain-2	Smart Security Systems	Cyber Security Fundamentals	Cryptography & Cyber Security	IoT-Based Data Analytics for Health Care	Blockchain Technology	Business Intelligence & Analytics
3	Domain-3	Interactive Digital Media	Interactive Art and Creative Coding	Gaming Design Fundamentals	Game Development Using Unity/ Game Development Using Godot	AR/VR & Game-Theoretic/ Algorithmic Game Theory	Unreal Engine Game Development for Beginners
4	Domain-4	Quantum Computing	Introduction to Quantum Computing	Quantum Algorithms	Quantum Machine Learning	Quantum Artificial Intelligence	Quantum Cryptography
5	Domain-5	Technology Innovation & Entrepreneurship	Product Strategy & Management	Business Management using Cloud	Business-to-Business Marketing	Sustainable Innovation Business	Digital Business

LINEAR ALGEBRA & DATA FOUNDATIONS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I /II

Subject Code	:		**Credits**	: 03
Hours /Week	:	03 Hours	**Total Hours**	: 45 Hrs L
L–T–P–J	:	3–0–0-0		

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamental principles of linear algebra and vector spaces used in computational systems.
2. Apply matrix operations and linear transformations for modelling engineering and AI problems.
3. Analyze uncertainty and randomness using probability theory and statistical methods.
4. Apply data preprocessing and regression techniques for predictive modelling.
5. Use mathematical tools for data-driven decision making in intelligent engineering systems.

BROAD COURSE DESCRIPTION

To develop the ability to analyze engineering problems using linear algebra, probability, and data analytics concepts, and to apply these mathematical foundations in building intelligent systems and AI-driven solutions responsibly

TEACHING-LEARNING PROCESS (GENERAL INSTRUCTIONS)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. Lecture-based conceptual teaching with illustrative examples.
2. Problem-solving sessions after each topic.
3. Use of MATLAB/Python for computational verification.
4. Group discussions and peer learning.
5. Activity-based and application-oriented learning.
6. Continuous internal assessment through quizzes and assignments.
7. Real-world case studies in probability and regression.

UNIT – I	09 Hours

SYSTEMS OF LINEAR EQUATIONS

Topics: System of Linear equations, Row reduction and echelon form, Rank of a matrix by row echelon form, Gauss elimination method, Inverse of a matrix using Gauss-Jordan method, LU decomposition.

Applications: Engineering system modelling, Circuit analysis, Computer graphics transformations, Robotics control systems.

UNIT – II	07 Hours

VECTOR SPACES

Topics: Vector spaces, Subspaces, Linear independence, Span, Bases and dimensions, Finite dimensional vector spaces.

Applications: Signal processing, Computer graphics, Feature representation in data science, Image compression techniques.

UNIT – III	10 Hours

LINEAR TRANSFORMATIONS AND EIGEN VALUE

Topics: Linear transformations, Matrices of linear transformations, Vector space of linear transformations, Inner product spaces, Orthogonal vectors, Projections, Gram-Schmidt orthogonalization process, Eigenvalues and eigenvectors and interpretation.

Applications: Image processing, Computer vision systems, Signal filtering, Structural engineering analysis.

UNIT – IV	09 Hours

PROBABILITY AND RANDOM VARIABLES

Topics: Random experiments, Algebra of events, Empirical basis of probability, Laws of probability, Conditional probability, Independence of events, Random variables (discrete and continuous), Mean, Variance, Standard deviation, Sum of independent random variables, Applications of probability to business and economics.

Applications: Risk analysis, Financial forecasting, Decision making under uncertainty, Reliability engineering.

UNIT – V	09 Hours

DATA ANALYTICS AND REGRESSION

Topics: Introduction to data, Data types and collection, Data preprocessing, Features of vectors, Multidimensional data representation, Dataset dimensionality, Simple regression models with one independent variable, Assumptions of regression models, Estimation of parameters, Standard error of estimator.

Applications Business analytics, Demand forecasting, Market trend analysis, Quality control systems.

LAB EXERCISES

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Systems of Linear Equations (03 Hours)					
1	Basic of Linear Algebra and Solving Methods	Solve systems of linear equations using Gauss Elimination.	MATLAB/PYTHON	1 h	CO1
2	Matrix Inverse & LU Decomposition	Compute matrix inverse and perform LU decomposition	MATLAB/PYTHON	1 h	CO1
3	Vectorized Matrix Computation	Implement matrix multiplication to simulate a single-layer neural network forward pass. Perform batch matrix operations to understand vectorized AI computation.	MATLAB/PYTHON	1 h	CO1
MODULE II: Exercises 4–6: Vector Spaces (03 Hours)					
4	Linear Independence & Basis	Verify linear independence of feature vectors and Compute basis and dimension of datasets	MATLAB/PYTHON	1 h	CO2
5	Vector Space Operations	Implement vector space operations (span, subspace verification).	MATLAB/PYTHON	1 h	CO2

6	Feature Space Visualization	Represent datasets as vectors in multidimensional space and Visualize 2D and 3D feature spaces.	MATLAB/PYTHON	1 h	CO2
MODULE III: Exercises 7–9: Linear Transformations and Eigen Value (06 Hours)					
7	Eigen Computation	Compute eigenvalues and eigenvectors.	MATLAB/PYTHON	1 h	CO3
8	PCA Implementation	Implement Principal Component Analysis (PCA) from scratch and Apply PCA on real dataset.	MATLAB/PYTHON	1 h	CO3
9	Orthogonalization & Projection	Implement Gram-Schmidt Orthogonalization. Perform vector projection and interpret results.	MATLAB/PYTHON	1 h	CO3
MODULE IV: Exercises 10–12: Probability and Random Variables (06 Hours)					
10	Statistical Measures	Simulate random experiments. Compute mean, variance, and standard deviation of datasets.	MATLAB/PYTHON	1 h	CO4
11	Naïve Bayes Classifier	Implement Naïve Bayes classifier.	MATLAB/PYTHON	1 h	CO4
12	Random Experiments	Generate discrete and continuous random variables.	MATLAB/PYTHON	1 h	CO4
MODULE V: Exercises 13–15: Transaction Management (06 Hours)					
13	Data Preprocessing	Load and preprocess dataset. Handle missing values and normalize data.	MATLAB/PYTHON	1 h	CO5
14	Linear Regression	Implement Simple Linear Regression from scratch.	MATLAB/PYTHON	1 h	CO5
15	Error Analysis & Visualization	Compute standard error and residual analysis. Visualize regression results.	MATLAB/PYTHON	1 h	CO5

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration

- Matrix computations used in neural networks
- Linear systems used in machine learning optimization
- Feature vector representation in machine learning
- Data compression and noise reduction
- Data preprocessing for machine learning and AI models

Authentic Intelligence Reflection:

1. Importance of verifying computational outputs in AI models
2. Responsible interpretation of data representations
3. Interpreting AI dimensionality reduction results responsibly
4. Understanding uncertainty in AI predictions
5. Importance of transparent data processing and model interpretation

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Apply matrix operations and solve systems of linear equations to perform computations used in AI models such as neural networks.	L3
CO2	Analyse vector spaces, basis, and linear transformations to understand feature representation and dimensionality reduction techniques in AI.	L4
CO3	Compute eigenvalues and eigenvectors and apply them in AI techniques like Principal Component Analysis (PCA), image compression, and pattern recognition.	L3
CO4	Apply probability laws and random variable concepts to develop probabilistic AI models such as Naïve Bayes classifiers and predictive systems.	L3
CO5	Perform data preprocessing and implement simple regression models to build foundational supervised learning models for AI-based decision-making.	L5

MAPPING LEVELS OF COS TO POS / PSOS

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO-1	3	3	2	2	1	1			1	2		3	2
CO-2	3	3	2	2	1				1	2		3	2
CO-3	3	3	3	2	2	1			1	2		3	3
CO-4	3	2	2	3	3	1			1	2	1	2	3
CO-5	3	3	3	3	3	2	1		2	2	1	3	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

RECOMMENDED TOOLS

1. Python
2. NumPy
3. Pandas
4. Scikit-learn
5. MATLAB
6. Data visualization tools

TEXT BOOKS:

1. Howard Anton & Chris Rorres, *Elementary Linear Algebra with Applications*, Wiley.
2. Gilbert Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press.
3. Sheldon Ross, *A First Course in Probability*, Pearson.
4. Douglas C. Montgomery & George C. Runger, *Applied Statistics and Probability for Engineers*, Wiley.

REFERENCE BOOKS:

1. David C. Lay, *Linear Algebra and Its Applications*, Pearson.
2. Erwin Kreyszig, *Advanced Engineering Mathematics*, Wiley.
3. William Feller, *An Introduction to Probability Theory and Its Applications*, Wiley.
4. James, Witten, Hastie & Tibshirani, *An Introduction to Statistical Learning*, Springer.

****

Engineering Physics

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

SubjectCode	:		**Credits**	: 04
Hours /Week	:	03 Hours	**Total Hours**	: 60 Hrs (45Hrs L +15Hrs P)
L–T–P-J	:	3–0–2-0		

Prerequisites: Basic knowledge of classical mechanics, wave theory, and limitations of classical physics to understand the need for quantum mechanics. Familiarity with mathematical tools such as calculus, differential equations, probability, and basic linear algebra for derivations and problem-solving. Introductory understanding of modern physics concepts to support topics like semiconductors, lasers, devices, and nanoscience.

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamental principles of quantum mechanics and apply them to engineering problems.
2. Explore quantum information and quantum computing concepts using quantum simulation tools.
3. Learn principles of quantum optics, lasers, and optical fibre systems and their role in data transmission.
4. Comprehend band structure of solids and analyse semiconductor device applications foundational to AI chip design. Understand thin-film fabrication and nanomaterial synthesis with applications in engineering, medicine, and AI sensors.
5. Use AI/ML tools (Python, MATLAB, PhET) to simulate physical systems and interpret experimental data. Understand how physics based LLM work.

BROAD COURSE DESCRIPTION

The course provides a comprehensive introduction to fundamental principles of engineering physics, including quantum mechanics, quantum information, optics, semiconductor physics, and nanotechnology. It emphasizes the application of these concepts to modern engineering systems, particularly in computing and electronics. With an integrated approach, the course incorporates AI and computational tools such as Python, simulation platforms, and machine learning techniques to model, analyse, and interpret physical phenomena. This enables students to connect theoretical physics with real-world applications in areas like quantum computing, AI hardware, optical communication, and advanced materials. Overall, the course is designed to develop analytical thinking, problem-solving skills, and interdisciplinary understanding, preparing students to contribute to emerging technologies at the intersection of physics and artificial intelligence.

UNIT – I	**08 Hours**

Failure of Classical Mechanics, Planck's hypothesis, Wave-Particle Duality, Uncertainty Principle, Wave function, Eigen functions and Eigen values, Schrodinger Time-Independent Wave Equation in 1D (Derivation), Particle in a Potential Well (Derivation), Numericals.

UNIT – II	**06 Hours**

Introduction to Quantum Information, Qubit, Bloch Sphere, Superposition, Entanglement, Quantum Gates, Quantum Measurement, Quantum Parallelism, Quantum Supremacy. Applications: Quantum Key Distribution (QKD), Quantum Sensors.

UNIT – III	**09 Hours**

Quantum Optics: Absorption, spontaneous and stimulated emission, Einstein coefficients, pumping, population inversion, resonators, laser characteristics, Nd:YAG and semiconductor lasers, engineering applications.

Fibre Communication: Optical fibre basics, numerical aperture, acceptance angle, step-index and graded-index fibres, fibre losses, applications.	
UNIT – IV	**14 Hours**
Semiconductor Physics: Classification of solids, band structure, Fermi level (intrinsic and extrinsic), Fermi-Dirac distribution, electron concentration in conduction band (Derivation), intrinsic carrier concentration, conductivity, Hall effect, Hall Coefficient (Derivation), quantum Hall effect (intro), Numericals. Semiconductor Devices: Zener diode, LED, Photodiode, Solar Cell, BJT. Thin Films: Introduction, advantages over bulk materials, PVD — thermal evaporation and sputtering, applications. Nano Science & Technology: Nanomaterials, classification, size-dependent properties, quantum confinement, top-down and bottom-up approaches (ball milling, photolithography), applications in medicine and engineering.	
UNIT – V	**08 Hours**
Probability Distributions in Physics & ML (2 Hour), Boltzmann Distribution (2 Hour), Energy Landscapes (1 Hour), Magnetohydrodynamics (2 Hour), Scaling and Nondimensionalizing (3 Hours).	

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

- **Module-1:** The integration of computational and interactive tools significantly improves the understanding of quantum mechanics concepts. Python can be used to generate Planck's spectral radiance plots and visualize probability densities ($|\psi|^2$) for systems like the particle-in-a-box. PhET simulations such as *Quantum Wave Interference* and *Quantum Bound States* provide intuitive insights into wave behavior and energy quantization. AI-assisted derivations and Python-based numerical solutions of the Schrödinger equation enhance analytical understanding. Additionally, AI-generated problem sets with numerical answer checkers support effective practice and learning.
- **Module-2:** IBM Qiskit or Quantum-circuit tools enables the creation and visualization of qubit states, including Bell state circuits and interactive Bloch sphere representations. It allows users to build quantum gate circuits and compare them with classical logic gates used in computer science. Real-world applications, including Quantum Key Distribution (QKD) for secure communication and quantum sensors in MRI and navigation, connect theory with practice. Overall, Qiskit serves as a comprehensive tool for both learning and applying quantum computing concepts.
- **Module-3:** PhET's *Lasers* simulation illustrates population inversion step by step. Gaussian beam propagation can be visualized using Python, with applications in AI-driven laser-guided surgery and manufacturing. Real-world examples such as VCSEL-based LiDAR and IR laser-based facial recognition highlight practical relevance. In fibre optics, tools like NA calculations, attenuation plots, and PhET's *Bending Light* simulation help compare fibre types and understand AI-managed networks.
- **Module-4:** Virtual lab simulations of Zener diode and LED I–V characteristics, along with Python-based solar cell efficiency optimization, connect theory to applications. Simulation of thermal evaporation combined with AI-based quality control using computer vision enhances understanding of PVD manufacturing processes. Case studies such as AI-optimized sputtering in semiconductor fabrication (e.g., TSMC, Intel) encourage critical student discussion. Python can be used to analyze quantum confinement energy as a function of particle size, along with AI-driven material property prediction using tools like the Materials Project API. Additionally, animations of photolithography and applications like AI-guided drug delivery nanoparticles support conceptual learning and student presentations.
- **Module-5:** Concepts such as data distributions, likelihood estimation, probabilistic models, and sampling-based learning form the foundation of modern AI techniques, including Boltzmann machines and softmax-based interpretations. Understanding neural network loss surfaces, transformer optimization, and training instability provides insight into how complex models learn and converge.

These ideas parallel physical systems, where relationships among velocity, pressure, temperature, and density in fluids are governed by fundamental principles balancing forces like inertia, viscosity, and gravity. Overall, this approach emphasizes system comparison, simplification, and gaining physical insight across disciplines.

COURSE OUTCOMES (CO)

CO	Course Outcome	Bloom's Level
CO1	Illustrate characteristics of semiconductor materials and devices	Apply (L3)
CO2	Demonstrate optical principles of diffraction and interference	Apply (L3)
CO3	Demonstrate frequency response curve for an LCR circuit	Apply (L3)
CO4	Illustrate characteristics of a dielectric material	Apply (L3)
CO5	Illustrate the characteristics of Qbit and manipulation of Qbit using quantum circuit	Apply (L3)

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	3	-	-	-	-	-	-	-	-	2	1	1
CO2	3	3	-	-	-	-	-	-	-	-	2	1	1
CO3	3	3	-	-	-	-	-	-	-	-	2	1	2
CO4	3	2	-	-	-	-	-	-	-	-	2	1	1
CO5	3	3	-	-	-	-	-	-	-	-	3	1	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

ENGINEERING PHYSICS LABORATORY

List of Experiments

No.	Experiment	Module Link
1	I-V Characteristics of Zener Diode	Module IV
2	Four Probe Technique — Resistivity	Module IV
3	Dielectric Constant Determination	Module IV
4	Fibre Optic System — NA, Acceptance Angle & Losses	Module III
5	Band Gap Energy of Intrinsic Semiconductor	Module IV
6	Diffraction Grating — Wavelength of Laser Light	Module III
7	Planck's Constant using LED	Module IV
8	LCR Series & Parallel Resonance	Module IV
9	Transistor Characteristics (NPN-CE)	Module IV
10	Hall Effect — Concentration, Mobility & Conductivity	Module IV
11	Realization of the hadamard gate and its geometric representation on the bloch sphere via quantum-circuit computing platform	Module II

12	Realization of the c-not gate and its bloch sphere visualization using	Module II

AI MINI-PROJECT GUIDELINE

Requirements

1. Teams of 3-4 students; one project per team
2. Choose a physics topic from Modules I–V and identify a real CSE/AI-linked problem
3. Apply at least one AI/computational tool (Python, Qiskit, MATLAB, or ML model)
4. Submit: Code/notebook + written report (1500-2000 words) + 10-minute presentation
5. Evaluated on: Physics understanding (40%), AI implementation (30%), Presentation (20%), Report quality (10%)

Suggested Project Ideas

1. **Quantum Circuit Simulator for Grover's Search — Qiskit (Module II)**
    - *Physics: superposition, quantum gates, measurement*
2. **AI-Assisted Optical Fibre Loss Characterisation — Python ML on lab data (Module III)**
    - *Physics: attenuation, NA, dispersion*
3. **Machine Learning on Semiconductor I-V Data — scikit-learn regression (Module IV)**
    - *Physics: Zener / solar cell I-V characteristics*
4. **Nanomaterial Property Predictor using Materials Project API (Module V)**
    - *Physics: quantum confinement, band gap*
5. **Schrodinger Equation Solver for Custom Potential Wells — Python finite-difference (Module I)**
    - *Physics: TISE, energy eigenvalues*

ASSESSMENT SCHEME

Professional Core Courses (Theory)

Criteria	Weightage of Marks
Lab Reports (experiments)	15
Class Tests / Quizzes	10
Mid-Semester Examination	30
AI Mini-Project (Group)	5
End-Semester Examination	40
Total	**100 marks**

PRIMARY TEXTBOOKS

1. Nielsen, M.A. & Chuang, I.L. — Quantum Computation and Quantum Information, Cambridge Univ. Press, 10th Anniversary Ed.
2. Thyagarajan, K. & Ghatak, A.K. — Lasers: Fundamentals and Applications, 2nd Ed., Springer India
3. Keiser, G. — Optical Fiber Communications, 5th Ed., McGraw-Hill
4. Kasap, S.O. — Principles of Electronic Materials and Devices, 4th Ed., McGraw-Hill
5. Beiser, A. — Concepts of Modern Physics, 6th Ed., McGraw-Hill
6. Engineering Physics Laboratory Manual, Dept. of Physics, DSU — Internal Publication, 2025

ADDITIONAL RESOURCES

1. IBM Qiskit Textbook — Free at qiskit.org/learn
2. NPTEL Lectures: Quantum Computing, Semiconductor Physics & Devices — nptel.ac.in

3. Google Colab Python notebooks — colab.research.google.com
4. PhET Simulations — phet.colorado.edu

5. Materials Project database — materialsproject.org
6. Goodfellow et al. — Deep Learning — Free at deeplearningbook.org
7. NVIDIA Deep Learning Institute (DLI) — Recommended Learning Paths for Module VI: (1) Fundamentals of Deep Learning — learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-FX-01+V3; (2) Physics-Informed Machine, Learning learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-OV-12+V1; (3) Simulation and Modelling with NVIDIA Modulus (PhysicsNeMo) — learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-OV-06+V1. Note: DLI content is recommended for self-paced supplementary study and should be selected to complement the specific topics covered in each session of Module VI. Instructors are encouraged to map each DLI module to the corresponding lecture topic before student enrolment to ensure curriculum alignment.

AI-POWERED ASSIGNMENTS

Assignment 1: AI-Generated Quantum Wave Function Analysis

1. Students prompt an AI tool: "Generate the wave function, probability density, and energy levels for a particle in a 1D infinite potential well."
2. Students must: analyze the correctness of boundary conditions and normalization
   i. identify errors in probability density ($|\psi|^2$) interpretation
   ii. improve the solution with proper derivation and graphical representation
3. Assessment focuses on conceptual clarity, mathematical accuracy, and quantum interpretation

Assignment 2: AI-Assisted Schrödinger Equation Derivation Review

1. Students prompt an AI tool: "Derive the Time-Independent Schrödinger Equation in one dimension."
2. Students must: verify the logical flow and physical assumptions used identify missing steps or incorrect reasoning refine the derivation with proper justification and physical meaning
3. Assessment focuses on derivation skills, conceptual understanding, and critical evaluation.

Assignment 3: AI-Based Quantum Circuit and Qubit Validation

1. Students prompt an AI tool: "Create a Bell state circuit and explain qubit superposition and entanglement using Bloch sphere representation."
2. Students must: evaluate correctness of circuit design and quantum states identify misconceptions in entanglement or measurement improve explanation with correct visualization and quantum logic
3. Assessment focuses on quantum computing understanding and analytical reasoning.

Assignment 4: AI-Supported Laser Physics and Einstein Coefficient Analysis

1. Students prompt an AI tool: "Explain population inversion and derive the relation between Einstein coefficients."

2. Students must: assess correctness of derivation and assumptions identify missing physical interpretation or errors enhance the explanation with diagrams and real-world laser applications
3. Assessment focuses on physical insight and application-oriented thinking.

Assignment 5: AI-Generated Semiconductor Physics Problem Evaluation

1. Students prompt an AI tool: "Calculate carrier concentration and Fermi level position for intrinsic semiconductors at different temperatures."
2. Students must evaluate correctness of formulas and units identify errors in temperature or doping dependence refine solution using proper equations and graphical interpretation.
3. Assessment focuses on numerical accuracy and semiconductor concepts.

Assignment 6: AI-Assisted Device Characteristics and Optimization

1. Students prompt an AI tool: "Generate I-V characteristics of a Zener diode and calculate solar cell efficiency under given conditions."
2. Students must: analyze correctness of graphs and operating regions identify unrealistic assumptions or incorrect parameters improve results with proper modeling and practical constraints.
3. Assessment focuses on device understanding and engineering application.

Assignment 7: AI-Based Nanotechnology and Quantum Confinement Study

1. Students prompt an AI tool: "Explain quantum confinement and calculate energy variation with particle size for nanomaterials."
2. Students must: verify mathematical relationship and trends identify simplifications or incorrect scaling behavior enhance explanation with correct models and applications.
3. Assessment focuses on nanoscale physics and analytical thinking.

Assignment 8: AI-Supported Fibre Optics and Communication Analysis

1. Students prompt an AI tool: "Calculate numerical aperture, acceptance angle, and attenuation in optical fibres."
2. Students must: evaluate correctness of equations and results identify missing parameters or assumptions improve solution with comparison of fibre types and real-world data.
3. Assessment focuses on problem-solving and communication systems understanding.

Assignment 9: AI-Based Probability and Boltzmann Distribution Evaluation

1. Students prompt an AI tool: "Derive and plot the Boltzmann distribution for a physical system."
2. Students must: verify correctness of derivation and normalization identify conceptual or graphical errors refine the solution with physical interpretation and applications.
3. Assessment focuses on statistical physics understanding and modeling.

Assignment 10: AI-Driven Interdisciplinary Physics Model Review

1. Students prompt an AI tool: "Explain how AI models relate to physical systems like fluid flow (velocity, pressure, temperature relationships)."
2. Students must: analyze correctness of analogies between AI and physics identify oversimplifications or incorrect mappings improve explanation with proper physical laws and equations.
3. Assessment focuses on interdisciplinary thinking and conceptual depth.

****

Computational Thinking & Programming using Python

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

SubjectCode :		**Credits**	: 03
Hours /Week :	04 Hours	**Total Hours**	: 45 Hrs (30Hrs L +15Hrs P)
L–T–P-J :	2–0–2–0		

Prerequisite: Fundamental computer knowledge

Course Learning Objectives:

This Course will enable students to:

1. Gain knowledge of fundamental programming concepts such as variables, expressions and basic constructs in Python.
2. Make use of functions, modules, and strings to develop Python programs for data processing.
3. Gain skills to develop programs using core data structures, file handling techniques, and standard Python libraries.
4. Understand object-oriented programming concepts including classes, objects, inheritance, polymorphism.
5. Make use of operator overloading and exception handling techniques to develop robust Python applications.

BROAD COURSE DESCRIPTION

This course introduces computational thinking and problem-solving using Python programming. It covers fundamental concepts such as control structures, functions, strings, data structures, and object-oriented programming with an emphasis on modular and efficient coding. The course incorporates the use of standard Python libraries for data handling and processing, along with file handling and exception handling to develop reliable programs. Students gain practical experience through laboratory exercises focused on real-world problem solving. The course also introduces foundational artificial intelligence concepts such as data preprocessing and basic pattern identification using Python libraries, along with awareness of modern computing paradigms such as accelerated computing for handling large-scale data applications. The course prepares students to design and implement structured Python-based solutions.

UNIT – I	06 Hours

PYTHON FUNDAMENTALS AND CONTROL STRUCTURES

Topics: Features of Python, variables, data types, operators, input/output handling, indentation and comments, type conversions, conditional statements, looping constructs, nested loops, break, continue, pass, use of else with loops.

Engineering Applications: Basic engineering calculator, Temperature alert system and Traffic signal control system using conditional statements and loops

UNIT – II	06 Hours

FUNCTIONS, MODULES AND STRINGS

Topics: Need for functions, function definition and function call, scope and lifetime of variables, return statement, lambda functions, recursive functions, modules and their usage, string operations, immutability, string formatting, built-in string methods, string slicing, membership operators, comparison of strings, and iteration over strings, introduction to regular expressions (re module) for basic pattern matching and text validation.

Engineering Applications: Log file analyzer for system monitoring and error detection, Text data validation and cleaning for healthcare or retail data

UNIT – III	07 Hours

DATA STRUCTURES IN PYTHON, FILE HANDLING AND LIBRARIES

Topics: Lists, tuples, sets, and dictionaries, operations on data structures, file handling techniques including file paths, file types, file operations (read, write, append), file positions, and file management (rename and delete), use of standard Python libraries: *math, random, datetime, NumPy, Pandas,* and *Matplotlib* for data processing and visualization, introductory exposure to scikit-learn for basic data modeling workflows.

Engineering Applications: Student record management system for academic data storage and retrieval, CSV-based sales data analysis and visualization in retail systems, Patient dataset processing and pattern identification in healthcare systems

UNIT – IV	05 Hours

OBJECT-ORIENTED PROGRAMMING

Topics: Classes and objects, class methods and use of *self*, constructors and destructors, class variables, public and private data members, private methods, calling methods, static methods, inheritance, types of inheritance, method overriding, and polymorphism, use of dataclasses for simple and structured data representation

Engineering Applications: Banking system for account management and transactions, Inventory management system for retail operations, Smart city resource management system for monitoring utilities

UNIT – V	06 Hours

OPERATOR OVERLOADING AND EXCEPTION HANDLING

Topics: Operator overloading, implementation of operator overloading, reverse addition, method overriding, errors and exception handling, handling multiple exceptions, *else* and *finally* clauses, raising exceptions, built-in and user-defined exceptions, clean-up actions, use of logging module to record program execution events and error messages (INFO, WARNING, ERROR levels); introduction to warnings module for handling runtime warnings

Engineering Applications: File-based transaction system with exception handling for invalid operations, Matrix operations using operator overloading, User input validation for order processing systems with exception handling for incorrect entries.

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Apply Python fundamentals including variables, data types, conditionals and loops to develop simple programs.	L3
CO2	Apply functions, modules and string operations to develop modular Python programs.	L3
CO3	Analyze and manipulate data using Python data structures (lists, tuples, sets, dictionaries), file handling techniques and standard Python libraries: *math, random, datetime, NumPy, Pandas,* and *Matplotlib*.	L4
CO4	Implement programs using object-oriented principles, including classes, objects, inheritance, and polymorphism.	L3
CO5	Implement operator overloading and exception handling techniques to develop robust Python applications.	L3

STRENGTH OF CO MAPPING TO PO/PSOS:

Table: Mapping Levels of Cos to Pos / PSOs

COs	Program Outcomes (Pos)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	2	2	3		2		1				2		1
CO2	2	2	3		2		1				2		1
CO3	3	3	3	2	3		1				2	2	1
CO4	2	3	3		2		1				2	2	1
CO5	2	3	3	2	2		1				2	2	1
Avg	11/15	13/15	15/15	4/15	11/15		5/15				10/15	6/15	5/15

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

Module 1: Implement a simple rule-based temperature alert system

a. Take temperature input
b. Use conditional statements to compare with threshold
c. Use loops to continuously monitor values
d. Display alert messages based on conditions

Module 2: Implement a text validation system:

a. Accept input data (e.g., email or ID)
b. Use string operations or **re module** for pattern matching
c. Validate format and clean the input data
d. Display valid/invalid results

Module 3: Implement a simple data analysis system:

a. Load dataset using **Pandas**
b. Perform preprocessing (filtering, handling missing values)
c. Compute basic statistics
d. Visualize results using Matplotlib
e. Identify simple patterns or trends
f. Relate the data processing workflow to parallel execution and accelerated computing concepts used in GPU-based systems (conceptual exposure only)

Module 4: Implement a class-based system:

a. Define a class (e.g., BankAccount or Student)
b. Implement attributes and methods
c. Use inheritance for extending functionality
d. Simulate real-world operations

Module 5: Develop a fault-tolerant input system:

a. Accept user input
b. Validate using condition
c. Handle errors using exception handling

d. Use logging to record errors
e. Ensure reliable program execution without failures

Authentic Intelligence Reflection:

- **Module 1:** Analysis of how conditional statements and loops are used to implement rule-based decision systems, evaluation of program logic correctness, and reflection on how simple automated decisions are applied in real-world applications.
- **Module 2:** Evaluation of how functions and modular programming improve code reusability and efficiency, and analysis of how string processing and pattern validation support handling of real-world textual data.
- **Module 3:** Analysis of how data structures, file handling, and Python libraries are used to process and manage structured datasets, evaluation of data analysis results, and reflection on how patterns are identified from real-world data.
- **Module 4:** Analysis of how object-oriented programming models real-world systems, evaluation of class design for modularity and maintainability, and reflection on how structured design supports scalable software development.
- **Module 5:** Evaluation of the effectiveness of exception handling and operator overloading in building robust programs, analysis of failure scenarios and error conditions, and reflection on the importance of reliability and fault tolerance in real-world systems.

LABORATORY COMPONENT

The laboratory emphasizes hands-on implementation of Python programs for problem solving. Students will use IDLE, VS Code, Jupyter Notebook or Google Colab and standard libraries (such as *math, random, datetime, re, json, logging, NumPy, Pandas,* and *Matplotlib*) to develop, analyze, and validate solutions for data processing and computational tasks.

Lab Experiments

1. Develop a program for a temperature alert system that triggers warnings when temperature exceeds a threshold using the random module for simulated input.
2. Develop a traffic signal control system using conditional statements and loops.
3. Create a log file analyzer to detect errors and generate reports using re for pattern matching.
4. Build a text validation system for healthcare or retail data inputs using re for format validation.
5. Develop a program to perform file operations for storing and retrieving records using json for structured data storage.
6. Develop a shopping cart application using lists and dictionaries to add, remove, update items, and calculate total bill with discounts.
7. Develop a student record management system using lists and dictionaries with file handling to add, update, delete, and retrieve student records, and store data persistently.
8. Develop a program to analyze sales data using Pandas, compute total sales, identify top-selling products, and visualize trends using Matplotlib.
9. Perform CSV data analysis and visualization using Pandas and Matplotlib libraries to load data, clean and process records, compute summary statistics, and generate visual representations such as bar charts or line graphs.
10. Develop an object-oriented banking system to create and manage accounts, perform operations such as deposit, withdrawal, and balance inquiry, and implement transaction handling using classes and methods.
11. Implement matrix operations using operator overloading with optional use of NumPy for comparison.
12. Develop a robust input validation system with exception handling and use of logging for error tracking.
13. Develop an ATM system using functions and file handling to perform operations such as balance inquiry, deposit, and withdrawal with proper validation and exception handling.

AI-Integrated Lab Exercises (Indicative)

1. Develop a Python program to load a dataset (CSV file) using Pandas, perform preprocessing (handling missing values, sorting, filtering), and compute basic statistics (mean, max, min). Visualize the results using Matplotlib.
2. Develop a Python program using math and random libraries to: Generate random numbers, Perform mathematical operations (square root, logarithm, trigonometric functions), Simulate a simple scenario (e.g., random score generation and analysis)
3. Develop a Python program to load a dataset (e.g., student marks) using Pandas and classify records into categories such as *pass/fail* or *high/medium/low* based on predefined conditions. Visualize the classification results using Matplotlib.
4. Develop a Python program to use the datetime library to process timestamps and analyze time-based data (e.g., login records or transactions). Store and process the data using Pandas, and generate summary insights.

ASSESSMENT SCHEME

Definitions

1. **"Internal Assessment"** refers to the Continuous Internal Assessment (CIA). The weightage of this component is **60%**. It includes:
   a. One **Mid Semester Examination (MSE)** (Descriptive-based),
   b. Six **Continuous Assessment Tool (CAT)**: It may include: Surprise test (Computer-based), Quiz, Project based learning, Reflection Note on Guest Lecture, Industrial visit, E-course certification, Building models, Group discussion, Case study, Seminar, or Open Book Assignment
   c. One **AI-powered Assignment (APA):** Use of AI tools for problem-solving while simultaneously developing Authentic Intelligence through critical evaluation, ethical reasoning, and responsible decision-making.
2. **"External Assessment"** refers to the Semester End Examinations (SEE). The weightage of this component is **40%**. At the end of semester, the final examination is conducted to evaluate student's performance.

INTEGRATED PROFESSIONAL CORE COURSES (THEORY + PRACTICAL)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) every 2 weeks (Minimum 05 assessments)	10
Mid Semester Examination (MSE)	20
Certification	10
Lab Record	10
Lab Examination	10
Total	**60 marks**

RECOMMENDED TOOLS

1. **Programming:** Python (IDLE / VS Code / Jupyter Notebook / Google Colab)
2. **Data Handling & Visualization:** NumPy, pandas, matplotlib
3. **AI-Assisted Development Tools:** GitHub Copilot, ChatGPT, code analysis and debugging tools

PRIMARY TEXTBOOKS

1. *Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", Green Tea Press, 2nd Edition.*
2. *Fabio Nelli, "Python Data Analytics", Apress, 2018.*

REFERENCE BOOKS

1. *John V. Guttag, "Introduction to Computation and Programming Using Python", MIT Press, 3rd Edition, 2021.*
2. *Tony Gaddis, "Starting Out with Python", Pearson, 4th Edition, 2019.*
3. *Al Sweigart, "Automate the Boring Stuff with Python", No Starch Press.*
4. *Luciano Ramalho, "Fluent Python", O'Reilly Media.*
5. *Reema Thareja, "Python Programming: Using Problem Solving Approach", 2nd Edition, Oxford University Press, 2019.*

ADDITIONAL RESOURCES

1. Infosys Springboard
2. Python Crash Course (Coursera)
3. Python Specialization (Coursera)
4. Python for Applied Data Science & AI (Coursera)
5. Python Official Documentation

DLI COURSE MAPPED

Fundamentals of Accelerated Computing with CUDA Python
(https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-AC-02+V1)

DLI LEARNING PATH

Accelerated Computing

****

Foundations of AI & Authentic Intelligence

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 03
Hours/ Week :	03 Hours	**Total Hours**	: 45 Hrs L
L–T–P-J :	3–0–0-0		

Course Learning Objectives:

This Course will enable students to:

1. Understand and develop a strong foundation in artificial intelligence, including its history, core principles, types (narrow vs. general AI), and real-world applications.
2. Explore how intelligent agents operate, including search algorithms, state-space representation, and decision-making strategies.
3. Learn to represent knowledge using logical and probabilistic models and apply reasoning techniques to solve complex problems.
4. Understand introductory machine learning concepts and evaluate the importance of fairness, transparency, accountability, and authenticity in AI systems.
5. Examine the distinction between artificial and authentic intelligence, focusing on human cognition, trust in AI, and responsible deployment in society.

BROAD COURSE DESCRIPTION

This course introduces first-year engineering students to the fundamentals of Artificial Intelligence along with the concept of Authentic Intelligence, emphasizing human-centered and ethical decision-making. It covers core AI topics such as intelligent agents, search strategies, knowledge representation, and problem-solving techniques. The course integrates real-world applications including fintech systems, blockchain analytics, and AI-driven decision-making tools. Students are exposed to AI-assisted learning using tools like Python, Excel, and visualization platforms. A unique AI² pedagogy is adopted, where students not only use AI tools but also critically evaluate their outputs. The course highlights biases, risks, and limitations of AI systems in practical scenarios. It promotes problem-first learning by connecting concepts with real-world challenges. Research-based learning is incorporated through case studies and student presentations. The course also encourages ethical reasoning, sustainability thinking, and financial inclusion awareness. Overall, it prepares students for responsible participation in AI-driven ecosystems.

UNIT – I	09 Hours

INTRODUCTION AND SEARCHING DECISIONS

Introduction, Definition, Future of Artificial Intelligence, Characteristics, Typical Intelligent agents, Problem solving approach, Problem Solving methods, Search strategies, Uniformed and informed, Heuristics, Local search, Algorithm and optimization problems, Searching with partial observations Constraint satisfactory problems ,Constraint propagation ,Back tracking search ,Game playing Optimal decision.

Applications: Heuristics are intelligent estimation techniques that improve search efficiency by prioritizing promising solution paths, Responsible deployment of AI chatbots, systematic techniques.

UNIT – II	09 Hours

AI REASONING

Alpha beta pruning, Stochastic games, First order predicate logic, Prolog programming, Unification, Forward Chaining, Backward chaining, Resolution, Knowledge Representation, Ontological Engineering, Categories, Objects, Events, Mental Events, Mental Objects, Reasoning Systems, Reasoning with default information, Typical AI Problems.

Applications: Heuristics are intelligent estimation techniques that improve search efficiency by prioritizing promising solution paths, Responsible deployment of AI chatbots, systematic techniques.

UNIT – III	09 Hours

INTELLIGENT AGENTS

Architecture for intelligent agents, Agent communication, Negotiation, Bargaining, Argumentation, Agents, Trust, Reputation, Multi agent systems, AI applications, Language Models, Information Retrieval, Information extraction, Natural language processing, Machine translation, Speech recognition, Robot Hardware, Perception.

Applications: Heuristics are intelligent estimation techniques that improve search efficiency by prioritizing promising solution paths, Responsible deployment of AI chatbots, systematic techniques.

UNIT – IV	09 Hours

INTELLIGENCE

Concept and principles of Authentic Intelligence, Augmented Intelligence vs Artificial Intelligence, Human-centered AI design principles, Value alignment between AI systems and human values, Human cognition and intelligence augmentation, Human–AI collaboration models.

***Applications*:** Ethical design of intelligent assistants, Responsible deployment of AI chatbots

UNIT – V	09 Hours

AUTHENTIC INTELLIGENCE IN EMERGING INTELLIGENT SYSTEMS

AI for human creativity and innovation, AI-driven knowledge discovery and learning systems, Intelligent systems for adaptive learning and personalization, Generative AI systems and creative intelligence, Authentic Intelligence and human–AI partnership.

Applications: Generative AI for design, art, and content creation, Intelligent tutoring systems and personalized education

COURSE OUTCOME:

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course, the student will be able to:		
1	Evaluate the characteristics and advantages of different intelligent agents	L2
2	Identify the different search strategies in AI	L3
3	Apply to solve AI problems	L3
4	Creating software agents in different ways of designing software agents	L3
5	Develop the applications of AI & solve the problems	L3

Mapping Levels of COs to POs / PSOs

Os				Program Outcomes (POs)								SOs	
		2	3	4	5	6	7	8	9	10	11	12		
O1		2	1	-	2	-	-	2	1	-	-	2	1	
O2		3	1	1	2	-	-	-	-	-	-	2	2	
O3		3	-	1	3	-	-	-	-	-	-	2	1	
O4		2	3	1	3	-	-	1	2	1	-	2	2	
O5		3	3	2	3	2	1	2	2	2	1	3	2	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION:

AI Integration:

1. Apply AI-based search algorithms and heuristic techniques to solve structured and unstructured problem spaces efficiently.
2. Use AI techniques such as logic programming, inference mechanisms, and knowledge representation to model intelligent reasoning systems.
3. Design and implement intelligent agents and AI applications including NLP, information retrieval, and multi-agent systems.
4. Incorporate human-centered AI design principles to develop systems aligned with human values and cognitive processes.
5. Apply generative AI and intelligent systems to enhance creativity, learning, and adaptive decision-making.

Authentic Intelligence Reflection:

1. Critically evaluate algorithmic decisions and ensure appropriate human judgment in selecting optimal problem-solving strategies.
2. Assess the correctness, limitations, and interpretability of AI-driven reasoning in real-world decision-making contexts.
3. Evaluate trust, reliability, and ethical implications of AI agents interacting in dynamic and real-world environments.
4. Reflect on ethical considerations, value alignment, and responsible AI usage in human–AI collaboration.

5. Analyze the balance between human creativity and AI augmentation while ensuring ethical and responsible innovation.

RESEARCH INTEGRATION TEACHING

1. Paper Review: Students study recent AI research on agents, search, and reasoning.
2. Case Studies: Analyze real-world AI systems to link theory with practice.
3. Mini Presentation: Present short reviews on emerging AI topics.
4. AI-Assisted Research: Use AI tools for summarization and comparison of studies.
5. Critical Reflection: Identify research gaps, limitations, and ethical issues.

ASSESSMENT SCHEME :

1. "**Internal Assessment (IA)**" refers to the Continuous Internal Assessment (CIA) with a weightage of 60%. It includes:

   a. **Mid Semester Examination (MSE)**: One descriptive examination to evaluate conceptual understanding of AI fundamentals and problem-solving approaches.

   b. **Continuous Assessment Tools (CAT – Six Components)**: Includes activities such as quizzes, surprise tests, case studies, group discussions, project-based learning, seminars, industrial visits, reflection notes, model building, e-course certifications, or open book assignments.

   c. **AI-Powered Assignment (APA):** Students utilize AI tools for solving problems while demonstrating Authentic Intelligence through critical evaluation, validation of AI outputs, ethical reasoning, and responsible decision-making.

2. "**External Assessment (EA)**" refers to the Semester End Examination (SEE) with a weightage of 40%. This examination evaluates students' overall understanding of AI concepts, application skills, and their

ability to analyze and solve problems effectively.

A. PROFESSIONAL CORE COURSES (THEORY)

Criteria	Weightage of Marks
Continuous Internal Assessment every 2 weeks (Minimum 06 assessments)	20
Mid-Semester Examination (MSE)	30
AI-powered Assignment / Mini Project*	10
Total	60 marks

AI-POWERED ASSIGNMENTS :

Assignment 1: AI-Based Problem Solving using Search Algorithms

Students prompt an AI tool: "Solve a pathfinding problem using A or heuristic search*"

Students must:

1. Analyze how the algorithm arrives at the solution
2. Evaluate efficiency and optimality
3. Suggest improvements or alternative approaches

Assignment 2: AI-Based Logical Reasoning Analysis

Students prompt an AI tool:"Solve a problem using propositional or predicate logic."

Students must:

1. Verify correctness of logical reasoning
2. Identify missing assumptions or errors
3. Evaluate interpretability of the solution

Assignment 3: AI-Based Intelligent Agent Design

Students prompt an AI tool: "Design an intelligent agent for a real-world scenario (e.g., chatbot, traffic system)."

Students must:

1. Analyze agent structure and environment
2. Evaluate decision-making capability
3. Identify ethical and real-world limitations

Assignment 4: AI-Based NLP Application Evaluation

Students prompt an AI tool: "Summarize or translate a given text using NLP techniques."

Students must:

1. Evaluate accuracy and contextual understanding
2. Identify biases or incorrect interpretations
3. Suggest improvements for better performance

Assignment 5: AI-Based Ethical Decision-Making Case Study

Students prompt an AI tool: "Provide a solution for an ethical dilemma involving AI (e.g., autonomous systems)."

Students must:

1. Analyze ethical reasoning provided by AI
2. Identify potential risks or biases
3. Propose a human-centered and responsible solution

TEXT BOOKS:

1. Co-Intelligence: Living and Working with AI, E. Mollick, 2024, Penguin Publishing Group.
2. Augmented intelligence: the business power of human–machine collaboration, J. Hurwitz, H. Morris, C. Sidner, and D. Kirsch, 2019, Auerbach Publications.
3. The Cambridge handbook of responsible artificial intelligence: interdisciplinary perspectives, S. Voeneky, P. Kellmeyer, O. Mueller, and W. Burgard, eds., 2022, Cambridge University Press.
4. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach‖, Prentice Hall, Third Edition, 2009.

REFERENCE BOOKS:

1. Bratko, ―Prolog: Programming for Artificial Intelligence‖, Fourth edition, Addison- Wesley Educational Publishers Inc., 2011.
2. M. Tim Jones, ―Artificial Intelligence: A Systems Approach (Computer Science) ‖, Jones and Bartlett Publishers, Inc.; First Edition, 2008.
3. William F. Clocksin and Christopher S. Mellish, Programming in Prolog: Using the ISO Standard‖, Fifth Edition, Springer, 2003.

E-RESOURCES : (RECOMMENDED TOOLS)

1. ChatGPT / Gemini for ideation
2. Jupyter Notebook / Google Colab / AnacondaGoogle Vertex AI
3. NetworkX or draw.io
4. AutoML platforms
5. OpenAI Gym
6. UCI Machine Learning Repository
7. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-RX-02+V2

ACTIVITY BASED LEARNING (SUGGESTED ACTIVITIES IN CLASS)

. 1. Group discussion on different search strategies to identify the nearest Patterns.

2.Collaborative Activity is minor project development with a team of 4 students.

****

INTRODUCTION TO ELECTRICAL AND ELECTRONICS ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code	:		**Credits**	: 03
Hours / Week	:	03 Hours	**Total Hours**	: 45 Hrs L
L–T–P-J	:	3–0–0-0		

Course Learning Objectives:

This Course will enable students to:

1. To provide a comprehensive understanding of the fundamentals of power electronic converters and devices.
2. To understand electric vehicle architecture and propulsion systems.
3. To analyze the interaction between batteries, power electronics, and electric drives using modeling and simulation by adopting AI tools.
4. To study charging techniques, battery management systems, and energy recovery mechanisms by adopting AI tools for modeling and simulation.
5. To familiarize with current trends and challenges in electric vehicle development.

TEACHING-LEARNING PROCESS (GENERAL INSTRUCTIONS)

These are sample new pedagogical methods where teacher can use to accelerate the attainment of the various course outcomes.

1. **The lecture method** means it includes not only the traditional lecture method, but also different types of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching**: Adopt the **Active learning** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, ask at least three Higher order Thinking questions in the class.
6. Adopt **Problem Based Learning**, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information utilizing AI tools.
7. Show the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every **concept can be applied to the real world** - and when that's possible, it helps improve the students' understanding.

UNIT – I	09 Hours

ELECTRICAL CIRCUIT FUNDAMENTALS

Basic electrical quantities such as charge, voltage, current, resistance, power, and energy; Ohm's Law; Kirchhoff's Voltage and Current Laws (KVL and KCL); series and parallel resistor networks; voltage and current division rules; introduction to AC and DC waveforms including RMS, peak, and average values.

Applications: Battery sizing for portable electronics, LED current limiting circuits, household wiring load estimation

UNIT – II	09 Hours

ELECTRICAL MACHINES AND POWER SYSTEMS

Magnetic circuits including flux, flux density, magnetomotive force (MMF), and reluctance; working principle, types, and applications of transformers; working principle of DC generators and motors; basics of electrical power generation, transmission, and distribution; introduction to renewable energy sources.

Applications: Transformer design for mobile phone chargers and SMPS power supplies, DC motor control

in robotics and electric vehicles.	
UNIT – III	**09 Hours**
SEMICONDUCTOR DEVICES AND ITS APPLICATION Overview of diode construction; diode under no-bias, forward bias, and reverse bias conditions; half-wave and full-bridge rectifiers; construction and operation of npn and pnp BJTs; input and output characteristics of the common-base configuration; regions of transistor operation active, cut-off, and saturation; transistor as a switch. **Applications:** AC to DC power conversion in mobile chargers and SMPS, signal demodulation in AM radio receivers, transistor switching in microcontroller output drivers.	
UNIT – IV	**09 Hours**
OPERATIONAL AMPLIFIERS Op-amp symbols and terminals; modes of operation including single- ended, differential, and common mode; properties of ideal and practical op-amps; applications such as inverting amplifier, non-inverting amplifier, summing amplifier, differential amplifier, differentiator, and integrator. **Applications:** Sensor signal amplification in biomedical instrumentation, audio mixing consoles, PID controller implementation in embedded systems.	
UNIT – V	**09 Hours**
DIGITAL ELECTRONICS AND EMBEDDED APPLICATIONS Binary number system; logic gates and their truth tables—AND, OR, NOT, XOR, NAND, NOR; combination of logic gates and basic numerical operations; sequential logic circuits including JK and D Flip-Flops; application of Flip-Flops in 4-bit binary counters; overview of microcontrollers at the block diagram level. **Applications:** Digital arithmetic units in microprocessors, binary counter design for frequency dividers and timers, flip-flop based registers in CPU design.	

NVIDIA DLI COURSES (RELEVANT TO THIS MODULE):

Getting Started With AI on NVIDIA Jetson Nano. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-RX-02+V2

AI INTEGRATION:

1. LTSpice simulation of DC resistive networks; automated nodal equation solving using NumPy; data logging and V-I characteristic plotting using Matplotlib and Pandas.
2. Flux-AI for circuit design.

AUTHENTIC INTELLIGENCE:

1. Risks of over-relying on simulation without hardware validation; importance of measurement uncertainty and tolerance; ethical responsibility in accurate reporting of experimental data.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Demonstrate a clear understanding of the basic principles of electrical and electronic components, including resistors, diodes, transistors, operational amplifiers, logic gates, transformers and microcontrollers.	L2
CO2	Apply basic circuit law's and principles to analyze electrical circuits, including resistive networks, AC/DC waveforms, and basic transformer and DC machines	L3
CO3	Apply knowledge of semiconductor devices, such as diodes and transistors, to explain rectification and switching behavior in electronic circuits.	L3
CO4	Apply the principles of operational amplifiers to configure basic circuits such as inverting, non-inverting, and summing amplifiers.	L3
CO5	Demonstrate conceptual understanding of digital logic and microcontroller systems, including logic gate combinations, flip-flops, counters, and block diagram-level microcontroller architecture.	L2

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	-	-	-	2	1	1
CO2	3	3	-	-	-	-	-	-	-	-	2	1	2
CO3	3	3	-	-	-	-	-	-	-	-	2	1	1
CO4	3	3	-	-	-	-	-	-	-	-	2	1	1
CO5	3	2	-	-	-	-	-	-	-	-	2	1	1

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", 11th Ed, Pearson Education, 2013.
2. M. Moris. Mano and Michael D. Ciletti, "Digital Electronics", 4th Ed, Pearson Education, 2006.
3. D.P.Kothari and I.J. Nagrath, "Basic Electrical Engineering", 4th Edition, Tata McGraw Hill, 2019.
4. B.L. Theraja and A.K. Therja, "A textbook of electrical technology, Vol. I (Basic electrical Engineering)", S. Chand Publishing, 23rd Rev Ed, 2006.

REFERENCE BOOKS:

1. David A Bell, "Electronic Devices and Circuits", 5th Ed, Oxford university press, 2008.
2. Millman & Halkias, "Electronics Devices and Circuits", 2nd Ed, McGraw Hill, 2010.
3. William H Hayt and Jack E Kimberly and Steven M Durbin, "Engineering Circuit Analysis" 8th Edition

McGraw-Hill, 2013.

ALTERNATIVE NPTEL/SWAYAM COURSE:

1. NPTEL- https://nptel.ac.in/courses/122/106/122106025
2. NPTEL- https://nptel.ac.in/courses/108/108/108108076
3. Virtual Labs- http://vlabs.iitkgp.ac.in/be/

****

Engineering Communication & Technical Writing

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code	:	**Credits**	: 02
Hours / Week	: 03 Hours	**Total Hours**	: 30 Hrs L
L–T–P-J	: 2–0–0-0		

Prerequisites: Basic English language proficiency and fundamental digital literacy

Course Learning Objectives:

This Course will enable students to:

1. Develop lexical and grammatical competence through AI-assisted practice.
2. Enhance listening, speaking, reading, and writing skills specifically for professional contexts.
3. Optimize oral and written communication utilizing AI feedback loops.
4. Familiarize themselves with employability skills and modern job search techniques.
5. Inculcate critical thinking and research skills, with a focus on AI-supported research writing and referencing

BROAD COURSE DESCRIPTION

This course introduces professional communication and technical writing in the age of Artificial Intelligence. It develops language, sociolinguistic, discourse, and strategic competencies through grammar, vocabulary, technical writing, presentations, and workplace communication. Students learn to use AI tools for grammar improvement, presentation design, research writing, and documentation while strengthening essential soft skills such as teamwork, adaptability, and emotional intelligence. The course also enhances employability through resume writing, cover letters, interview preparation, and collaborative projects, enabling students to communicate technical ideas effectively and professionally in academic and workplace environments.

UNIT – I — **06 Hours**

LANGUAGE COMPETENCE AND AI TOOLS

Listening vs. hearing; Intensive/extensive reading; Pronunciation guidelines; Grammar and vocabulary building.

Applications: Everyday professional discussions, peer interactions, collaborative initiatives.

UNIT – II — **06 Hours**

ADVANCED FRAMEWORKS FOR PROFESSIONAL DISCOURSE

Communication skills and critical thinking; Presentation skills; Group discussion techniques.

Applications: Organizing ideas in group settings, overcoming expression inhibitions, improving nonverbal communication.

UNIT – III — **06 Hours**

ACADEMIC/RESEARCH/TECHNICAL WRITING

Narrowing research topics; Writing minutes, memos, and emails; Academic report writing; Referencing conventions; Technical paper writing.

Applications: Workplace documentation, professional email etiquette, legally bound reports.

UNIT – IV — **06 Hours**

EMPLOYABILITY SKILLS IN THE CONTEXT OF AI

Drafting CVs and resumes; Cover letters; Interview techniques; Professional etiquette.

Applications: Presenting achievements effectively, showcasing personality in job applications.

UNIT – V — **06 Hours**

WORKPLACE COMMUNICATION AND SOFT SKILLS

Emotional intelligence; Teamwork; Problem-solving through communication; Group project planning and

execution.
Applications: Team collaboration and maintaining human connections while using AI tools.

COURSE OUTCOMES (CO)

COs	Outcome	Bloom's Taxonomy Levels
CO1	Apply appropriate grammar and vocabulary in technical and professional communication; develop competence for presentations and group discussions.	L3
CO2	Demonstrate advanced professional discourse skills, including critical thinking and presentation design.	L4
CO3	Produce academic, research, and technical writing (emails, reports, papers) using AI tools for clarity and proper referencing.	L6
CO4	Use AI tools responsibly to enhance communication, research, and writing tasks.	L5
CO5	Write professional documents like resumes and cover letters while applying soft skills such as emotional intelligence and teamwork.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	-	-	-	-	-	-	-	-	2	1	1
CO2	3	3	-	-	-	-	-	-	-	-	2	1	2
CO3	3	3	-	-	-	-	-	-	-	-	2	1	1
CO4	3	3	-	-	-	-	-	-	-	-	2	1	1
CO5	3	2	-	-	-	-	-	-	-	-	2	1	1

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

- **Module-1:** The AI tools used to develop the language competence are Grammarly, QuillBot, BBC Learning English App ELSA Speak and Otter.ai, specifically for LSRW skills and grammatical competence.
- **Module-2:** ChatGPT (brainstorming), Canva AI (design), Kialo Edu (debates and Group discussions. Systematic approach using AI tools to optimize the communicative skills mainly focusing on the organising the concepts and ideas and context specific presentation.
- **Module-3:** Grammarly, Hemingway Editor, Explainpaper. Email etiquettes, modifying the written expressions/writing in different cultural, academic and professional contexts.
- **Module-4:** The tools like VMock AI, ChatGPT, Peergrade are used by students to familiarize with employability skills and to meet the requirement of their specific professional domain.
- **Module-5:** Role-play, Peergrade

Authentic Intelligence Reflection:

- **Module-1:** Revising grammar optimizes competence; pronunciation improves clarity; reading enhances thought coherence.
- **Module-2:** GDs build cohesive idea presentation, self-confidence, and general awareness and improve the communication skills of the students.
- **Module-3:** Email etiquette builds trust; technical writing tailor the content to audiences; ethics ensure appropriate referencing
- **Module-4:** Tailored resumes highlight specific skills; cover letters reveal formal personality.
- **Module-5:** Fosters emotional intelligence, teamwork, and efficient problem-solving for real-world projects.

RESEARCH INTEGRATION TEACHING

Students get familiarized with Multidisciplinary Research using AI Tools and formal academic writing through their Assignment project and reports.

ASSESSMENT SCHEME

Criteria	Weightage of Marks
Continuous Assessment	20
Computer based Test (CBT)	40
Mini Project and Written Assignment	40
Total	100 marks

RECOMMENDED TOOLS

Writing & Grammar: Grammarly, QuillBot, Hemingway Editor
Speech & Listening: ELSA Speak, Otter.ai, BBC Learning English App
Research & Design: Explainpaper, ChatGPT, Canva AI, Kialo Edu
Employability: V Mock AI, Peergrade

SUGGESTED READING

1. Dhanavel, S.P. "English and Communication Skills for Students of Science and Engineering". Orient Blackswan Pvt. Ltd., 2009.
2. Meenakshi Raman and Sangeetha Sharma. "Technical Communication- Principles and Practice". 3rd Edition, Oxford University Press, 2009.
3. Murphy R. "English Grammar in Use", Cambridge University Press, 2012.
4. N. Krishnaswamy and T. Sri Raman. "Creative English for communication", Macmillan Publication, 2005
5. Day, R.A. "Scientific English: A Guide for Scientists and Other Professionals". 2nd Edition, Universities Press, 2000.

E-RESOURCES:

- Technical Communication Book
- HKS Communications Slides
- YouTube:CommunicationSkills

ASSIGNMENTS

Assignment 1: Campus to Community: Needs analysis of local issues (e.g., waste management) followed by digital storytelling campaign
Community Immersion & Needs Analysis. The project provides an opportunity for the students to explore local communities to identify pressing issues (e.g., waste management, education gaps, elder care). Problem-Based Learning (PBL) Identify the problem, research solutions, and present actionable plans. Digital Storytelling & Campaigning involves Creating posters or social media campaigns/ posts to raise

awareness.

Assignment 2: – Clayconomy: **Sculpting Expression through Ideas &Innovations**

Mind mapping social issues and translating them into tangible products or patent proposals. Ideation & Theme Setting, Brainstorm themes, Mind Mapping (Students sketch ideas linking social issues to sculptural forms), Sculpting Expression includes the translation of ideas into tangible form/ product, the reflection & Exhibition, Presentation & Reflection Journals should be produced through PPT slides. The students need to create a Product/ App/ Presentation/ Writing a Proposal for Patenting.

For example

● A new product (e.g., eco-friendly drone, smart helmet)
● A new app (e.g., campus navigation, mental health tracker)
● A new process (e.g., automated attendance system, waste segregation
method) Materials- Clay/ colours/ papers, Area- Problem statement- Target users- Proposed Solution- Basic technical feasibility-Process description- Product

Assignment 3: Bumps, Breakthroughs & Becoming: The Plays That Shape Us

The students have to perform a specific scene live, using only the texts provided. Live performance of literary scenes to build confidence in communication and group coordination. Selected texts are: 4 Shakespearean plays, 2 Contemporary plays and 3 Plays from Indian literature in English. The permitted time for the Performance will be (5- 7 minutes). (Costumes optional)

Technical/Academic Research Report Writing for the Assignments 1/2/3

Students will follow the format given below for the written assignments.
Abstract, Objectives of the project, Research questions or guiding problem statements. Literature Review, (Brief overview of existing studies, theories, or practices related to the project). Methodology includes the approach taken, Data collection (surveys, interviews, observation, brainstorming). Tools used (AI tools, communication frameworks, design software). Steps followed in project execution need to be mentioned highlighting the relevance of the methods. Implementation and detailed description of how the project was carried out with the relevant diagrams, flow charts and visuals. Results and Discussion must include Presentation of findings, outcomes, or products of the project. Conclusion includes how the research can be taken forward. Project 3 involves the textual analysis of the chosen text. References and Appendices need to be included. (survey forms, scripts, campaign posters, or mind maps.

****

BIOLOGY FOR ENGINEERS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code	:		**Credits**	: 02
Hours / Week	:	03 Hours	**Total Hours**	: 30 Hrs L
L–T–P-J	:	2–0–0-0		

Prerequisites: Basic Biology

Course Learning Objectives:

This Course will enable students to:

1. Acquire an understanding on basic modern biological concepts with an emphasis on how bio-processes are analogous to engineering field, as a multidisciplinary field.
2. Develop an understanding of fundamental engineering principles as they operate within physiological systems, and apply this knowledge to engineering design and problem-solving using functional examples from the human body.
3. Explain how biologically derived solutions can serve as a foundation for designing and developing improved processes and products, ultimately contributing to enhanced quality of life.

BROAD COURSE DESCRIPTION

Biology for Engineers is a multidisciplinary course that bridges biological sciences with engineering and artificial intelligence. The course focuses on understanding how biological systems function and how these principles can be translated into engineering solutions.

Students will explore concepts such as biomimetics, bioenergy, biomechanics, bioelectronics, and biopharma, along with their real-world applications. The course integrates AI-driven approaches for modeling biological systems, analyzing complex datasets, and designing intelligent solutions.

A key focus is on **Authentic Intelligence**, where students critically evaluate AI outputs, combine human reasoning with machine intelligence, and ensure ethical applications in healthcare, biotechnology, and engineering systems.

By the end of the course, students will be able to apply biological inspiration and AI tools to solve real-world engineering challenges.

UNIT – I	**06 Hours**

BIOMIMETICS

Topics: Biomimetics & related branches (Bioinformatics, Bioelectronics, Biomechanics) – Biofluid-Blood, heart mechanics & blood pressure **-** Life molecules (water, carbohydrates, proteins, lipids, nucleic acids)**-** Biological data (genomics, proteomics, metabolomics)**-** Biomimetic AI & bio-inspired robotics

Engineering Applications:

- Healthcare (diagnosis, heart monitoring) - AI-driven modeling of biological systems for optimization
- Design of energy-efficient systems inspired by nature (e.g., bird flight → aerodynamics)
- Biological data analysis

UNIT – II	**06 Hours**

BIOENERGY

Topics: Unit of life: Human and Plant cell- Metabolism: Enzymes as Bio-catalysts and physiological entities, Anabolism- Bioenergy from Sun-Photosynthesis, catabolism, AI in Modelling metabolic and biochemical pathways

Engineering Applications:

- Energy production in living systems
- Photosynthesis in plants and food production
- Digestion and respiration in humans

- Medical and agricultural improvements

UNIT – III	06 Hours

BIOMECHANICS

Topics: Biomechanics: Human body movement mechanics - Force and vector of body movement - Movement angles and joint motion - Muscle contraction and relaxation - Posture: static and dynamic - Ideal vs abnormal posture - Practical movements: stepping, lifting, sitting, standing - AI-based smart prosthetic limbs

Engineering Applications:

- Improving posture and movement efficiency
- Injury prevention and rehabilitation
- Sports performance enhancement
- Design of ergonomic systems
- Development of prosthetic limbs

UNIT – IV	06 Hours

BIOELECTRONICS

Topics: Bioelectronics: Brain–computer interaction **-** Artificial Neural Networks (pattern & image recognition) **-** Biosensors and AIoT in healthcare **-** Bionic eye (mechanism of vision) **-** Electronic nose (bio-olfactory mechanism) **-** Nerve and cardiac impulses **-** Biological clock and circadian rhythm

Engineering Applications:

- Brain-computer interfaces in medical technology
- Disease diagnosis using biosensors
- Smart healthcare systems (AIoT)
- Vision restoration (bionic eye)
- Smell detection (electronic nose)
- Monitoring heart and nerve activity

UNIT – V	06 Hours

BIOPHARMA

Topics: Metabolic syndromes - Cancer and its diagnostics - Lab-on-a-chip technology - Drug development and discovery - AI in drug discovery and vaccine development.

Engineering Applications:

- Early diagnosis and monitoring of diseases
- Personalized medicine for metabolic disorders and cancer
- High-throughput screening in drug development
- Point-of-care testing with lab-on-a-chip devices

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	**Apply and utilize** essential knowledge of the biological mechanisms of living organisms from the perspective of engineers and find solutions to solve bio-engineering problems with appropriate tools	**L3**
CO2	**Distinguish and make use of** optimal designs in engineering that are bio-mechanical in nature and build and use by observing and understanding bio-physiological processes involved in sensing, locomotion, and knowledge application of range of bio-chemicals.	**L3**
CO3	**Demonstrate** that bio-chemical, bio-sensory, bio-processes could be path-finders to optimise similarities for functional aspects of electronic, computer, mechanical, electrical machines.	**L3**

CO	Outcome	Bloom's Taxonomy Level
CO4	**Understand and apply** electronic principles to biological systems for monitoring, diagnosis, and therapeutic intervention at cellular and physiological levels.	**L2**
CO5	**Develop and produce** safe, effective therapeutic products using biological systems for the prevention, diagnosis, and treatment of diseases.	**L4**

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	-	-	-	-	-	-	-	-	2	1	1
CO2	3	3	-	-	-	-	-	-	-	-	2	1	2
CO3	3	3	-	-	-	-	-	-	-	-	2	1	1
CO4	2	3	-	-	-	-	-	-	-	-	2	1	1
CO5	3	3	-	-	-	-	-	-	-	-	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

- **Module-1**: This module reflects authentic intelligence by helping students analyse biological datasets, predict molecular structures, model metabolic pathways and use biomimetic algorithms & robotics.
- **Module-2** This module reflects authentic intelligence by aiding students in Modelling metabolic pathways, predicting enzyme activity, analysing biochemical reactions, disease prediction and drug design.
- **Module-3**: This module reflects authentic intelligence by encouraging students to par-take in the creation of smart prosthetic limbs with adaptive movement, motion analysis and posture correction, predictive models for muscle and joint behaviour and rehabilitation using AI-assisted devices.
- **Module-4**: This module reflects authentic intelligence by promoting the development of Neural networks for image and pattern recognition, AI-based biosensors for real-time health monitoring, AIoT systems for smart healthcare, signal processing of nerve and cardiac impulses and modelling circadian rhythms and biological patterns
- **Module-5**: This module reflects authentic intelligence by emphasizing on predicting drug-target interactions, designing new drug molecules and vaccines, modeling metabolic pathways for therapy optimization, enhancing diagnostic accuracy and speed.

Authentic Intelligence Reflection:

- **Module-1**: This module reflects authentic intelligence by the integration of human understanding with AI in biology and engineering, helps interpret data from genomics, proteomics, and metabolic pathways, supports decision-making in biomimetics and bio-inspired systems, ensures ethical and effective use of AI in healthcare, biotechnology, and engineering.

- **Module-2**: This module reflects authentic intelligence by Combining human understanding with AI analysis, interpreting metabolic and biochemical data and making ethical and accurate biological decisions.
- **Module-3**: **:** This module reflects authentic intelligence by focusing on Combining human understanding with AI for movement analysis, interpreting body mechanics for better prosthetic design and ensuring safe and ethical use in healthcare and rehabilitation
- **Module-4**: This module reflects authentic intelligence by encouraging the design of AI in bioelectronic systems, interpreting neural and sensory data effectively and ensuring ethical use in healthcare technologies
- **Module-5** module reflects authentic intelligence by promoting the development of AI for drug and vaccine development, interpreting complex biological data responsibly and making informed decisions in therapy and diagnostics

RESEARCH INTEGRATION TEACHING

Students review at least one topic:

- AI in genomics and precision medicine
- AI-driven drug discovery
- Bio-inspired robotics
- AI in healthcare diagnostics

ASSESSMENT SCHEME

Definitions

1. **"Continuous Internal Assessment** (CIA)**"** with a weightage of **60%**. It includes:
   a. **Mid Semester Examination (MSE)**: One descriptive examination to evaluate conceptual understanding of AI fundamentals and problem-solving approaches.
   b. Continuous Assessment Tools (CAT – Six Components): Includes activities such as quizzes, surprise tests, case studies, group discussions, project-based learning, seminars, industrial visits, reflection notes, model building, e-course certifications, or open book assignments.

Professional Core Courses (Theory)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) – Quiz, Assignments, Write ups	60
Mid Semester Examination (MSE)	40
Total	**100 marks**

RECOMMENDED TOOLS

- **MATLAB** – modeling biomechanics, system simulations
- **ANSYS** – biomimetic design testing
- **PyMOL** PyMOL – protein structure visualization
- **MATLabVIEW** – biomedical signal acquisition
- **BioSPPY** – ECG/PPG analysis
- **TensorFlow** – disease prediction models
- **AlphaFold** – protein folding prediction
- **AutoDocK**– drug design
- **BLAST** – gene/protein comparison
- **Rosetta** – protein engineering
- **Python** – core tool for AI and bioinformatics
- **R (programming language)** – statistical biology

SUGGESTED READING:

1. "Biology for Engineers" by Arthur T. Johnson: published by CRC Press in 2010.
2. *Biology for Engineers*. Author *Dr. M. Srinivasan & Dr. Santosh Kumar*;Khanna Publishers. ISBN: 978-93-92549-01-4
3. Campbell, N. A.; Reece, J. B.; Urry, Lisa; Cain, M,L.; Wasserman, S. A.; Minorsky, P. V.; Jackson, R. B. Pearson. "Biology: A global approach", , Global Edition, 10/E, 2014
4. David Nelson, Michael Cox. "Lehninger Principles of Biochemistry". W H Freeman & Company, Seventh Edition, 2017.
5. Janine M Benvus. "Biomimicry: Innovation inspired by Nature". William Morrow Paperbacks, 2002.
6. John Holland (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
7. Melanie Mitchell (1998). *An Introduction to Genetic Algorithms*. MIT Press.
8. Lecture Notes, PPT slides by course instructor.

****

KANNADA KALI (ಕನ್ನಡ ಕಲಿ)

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code	:		**Credits**	: 01
Hours / Week	:	03 Hours	**Total Hours**	: 15 Hrs L
L–T–P-J	:	1–0–0-0		

Prerequisites: Basic understanding of regional languages and communication skills

Course Learning Objectives:

This Course will enable students to:

1. Understand the origin and development of Kannada language and Karnataka culture.
2. Learn Kannada alphabets, numbers, pronunciation, and vocabulary.
3. Develop basic grammar and sentence construction skills.
4. Communicate effectively in day-to-day situations using Kannada.
5. Utilize AI tools such as translators, chatbots, and voice assistants responsibly for language learning.

COURSE VISION

To enable non-Kannada learners to acquire foundational Kannada communication skills for academic, social, and cultural integration while promoting responsible usage of AI-enabled language learning tools and authentic communication abilities.

BROAD COURSE DESCRIPTION

Kannada Kali is an ability enhancement course designed for non-Kannada learners to develop foundational communication skills in Kannada language. The course introduces students to the origin, culture, alphabets, grammar, pronunciation, vocabulary, and conversational aspects of Kannada.

The course integrates AI-based tools such as translation systems, voice assistants, OCR systems, and NLP chatbots to enhance learning experiences. Simultaneously, it promotes Authentic Intelligence by encouraging students to critically evaluate AI-generated outputs and develop real-life communication skills independently.

The course also emphasizes practical learning through activity-based teaching methods, conversational practice, pronunciation exercises, and cultural interaction.

UNIT – I	03 Hours

INTRODUCTION TO KANNADA AND CULTURE

Topics: Origin and evolution of Kannada language – Karnataka geography and culture – Heritage and traditions – Importance of Kannada language – Cultural identity and social integration.

Applications:

- Cultural understanding and integration
- Communication readiness in Karnataka
- Social interaction in regional environments
- Language appreciation and heritage awareness

UNIT – II	03 Hours

BASICS OF KANNADA LANGUAGE

Topics: Kannada alphabets (Varnamale) – Swaragalu and Vyanjanagalu – Numbers – Basic pronunciation – Reading and writing practice.

Applications:

- Reading Kannada signboards and notices
- Writing simple Kannada words
- Basic pronunciation and listening skills
- AI-assisted pronunciation learning

UNIT – III	03 Hours

GRAMMAR AND VOCABULARY

Topics: Vocabulary building – Pronouns – Verbs – Tenses – Sentence formation – Everyday vocabulary – Basic grammar rules.

Applications:

- Constructing simple Kannada sentences
- Daily communication
- Understanding grammar usage
- Vocabulary enhancement for practical speaking

UNIT – IV	03 Hours

COMMUNICATION SKILLS AND CONVERSATION

Topics: Greetings – Self-introduction – Asking questions – Daily conversations – Phrase making – Colloquial Kannada communication.

Applications:

- Classroom communication
- Market and travel conversations
- Interpersonal communication
- Practical spoken Kannada usage

UNIT – V	03 Hours

AI TOOLS FOR LANGUAGE LEARNING

Topics: Google Translate – Voice Assistants – NLP Chatbots – OCR tools – AI-based pronunciation systems – Ethical use of AI tools.

Applications:

- AI-assisted language learning
- Pronunciation improvement
- Translation support
- OCR-based text learning
- Responsible AI usage in communication

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Students will be able to describe the origin of Kannada language and Karnataka culture.	L2
CO2	They will be able to identify Kannada alphabets, numbers, pronunciation patterns, and vocabulary.	L1
CO3	Students will gain the ability to apply grammar concepts such as pronouns, verbs, and tenses in sentence construction.	L3
CO4	They will be able to construct simple Kannada sentences and communicate in everyday situations.	L3
CO5	Students will be able to demonstrate conversational Kannada skills using appropriate vocabulary and cultural expressions.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	-	-	-	-	-	-	-	-	2	1	1

CO2	3	2	-	-	-	-	-	-	-	-	2	1	1
CO3	3	3	-	-	-	-	-	-	-	-	2	1	2
CO4	3	3	-	-	-	-	-	-	-	-	2	1	2
CO5	3	3	-	-	-	-	-	-	-	-	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

ASSESSMENT SCHEME

Definitions : Continuous Internal Assessment (CIA)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) – Viva and Assignment (AI)	25
Mid Semester Examination (CBT)	15
Attendance	10
Total	**50 Marks**

ACTIVITY BASED LEARNING (SUGGESTED ACTIVITIES IN CLASS)

- Alphabet writing and pronunciation practice
- Vocabulary games and quizzes
- Role-play conversations
- Listening and speaking exercises
- AI-assisted translation activities
- Group discussions in simple Kannada
- Sentence formation exercises
- Short presentations in Kannada
- Cultural story telling activities
- Reflection logs (AI vs real communication)

RECOMMENDED TOOLS

Language Learning Tools

- Google Translate
- Google Assistant
- OCR Applications

AI & NLP Tools

- ChatGPT
- NLP Chatbots
- Voice Recognition Systems

LEARNING PLATFORMS

- YouTube Kannada Learning Channels
- Duolingo
- Digital Kannada Learning Platforms

PRIMARY TEXTBOOKS

1. Kannada Kali – Dr. Lingadevaru Halemane
2. Kannada Paatagalu – Dr. Chandrashekara Kambara
3. Basic Grammar – SLN Sharma & K Shankaranarayana
4. Spoken Kannada – Kannada Sahitya Parishat

5.

ADDITIONAL RESOURCES

1. Kannada Learning Materials by Kannada Sahitya Parishat
2. Basic Conversational Kannada Guides
3. Digital Language Learning Resources
4. AI-based Language Learning Platforms

****

KANNADA MANASU (ಕನ್ನಡ ಮನಸು)

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 01
Hours / Week :	03 Hours	**Total Hours**	: 15 Hrs L
L–T–P-J :	1–0–0-0		

Prerequisites: Basic understanding of Kannada language, literature, and communication skills

Course Learning Objectives:

This Course will enable students to:

1. Understand Kannada language, culture, and literary heritage.
2. Develop appreciation towards Kannada literature and value-based learning.
3. Analyze social harmony, coexistence, and ethical themes in literary texts.
4. Apply Kannada communication skills in practical and real-life contexts.
5. Utilize AI tools for language and literature learning while maintaining authentic understanding.

COURSE VISION

To promote Kannada language, culture, literature, and ethical values among students through literary appreciation, communication skills, and responsible usage of AI-enabled learning tools while developing authentic understanding and cultural awareness.

BROAD COURSE DESCRIPTION

Kannada Manasu is an ability enhancement course aimed at promoting Kannada language, culture, literature, and ethical values among students. The course introduces learners to selected literary texts, moral values, social harmony, and cultural identity through Kannada literature.

The course integrates AI-based tools such as translation systems, OCR tools, voice assistants, and NLP chatbots to enhance understanding and interaction. Simultaneously, it promotes Authentic Intelligence by encouraging students to critically evaluate AI-generated outputs and develop independent interpretation skills.

The course also emphasizes communication, cultural awareness, literary appreciation, ethical learning, and practical Kannada writing skills.

UNIT – I	03 Hours

CULTURE (ಸಂಸ್ಕೃತಿ)

Topics: Kannada cultural heritage – Works of Purandaradasa – Chidananda Murthy writings – Karnataka traditions – Cultural identity and values.

APPLICATIONS:

- Understanding Karnataka culture and heritage
- Literary appreciation
- Cultural awareness and social integration
- Value-based learning

UNIT – II	03 Hours

VALUES (ಮೌಲ್ಯಗಳು)

Topics: Moral values in Kannada literature – Ethical reasoning – Folklore and poems – Life lessons through literature.

Applications:

- Ethical understanding
- Value-based thinking
- Personality development
- Social responsibility

UNIT – III	03 Hours

CO-EXISTENCE (ಸಹಬಾಳ್ವೆ) Topics: Social harmony – Human relationships – Coexistence in society – Literary interpretation of social themes. **Applications:** • Social awareness • Human values and relationships • Community understanding • Peaceful coexistence	
UNIT – IV	**03 Hours**
LANGUAGE & IDENTITY (ನಾಡು-ನುಡಿ ಚಿಂತನೆ) Topics: Kannada language pride – Linguistic identity – Classical and modern Kannada texts – Importance of regional language. **Applications:** • Language identity and preservation • Communication skills • Literary understanding • Regional language awareness	
UNIT – V	**03 Hours**
LETTER WRITING (ಪತ್ರಲೇಖನ) Topics: Formal letters – Informal letters – Letter writing formats – Practical Kannada writing skills. **Applications:** • Practical communication • Official letter writing • Personal communication skills • Written Kannada proficiency	

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Students will be able to describe the origin of Kannada language and Karnataka culture.	L2
CO2	They will be able to identify Kannada alphabets, numbers, pronunciation patterns, and vocabulary.	L2
CO3	Students will gain the ability to apply grammar concepts such as pronouns, verbs, and tenses in sentence construction.	L4
CO4	They will be able to construct simple Kannada sentences and communicate in everyday situations.	L3
CO5	Students will be able to demonstrate conversational Kannada skills using appropriate vocabulary and cultural expressions.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	-	-	-	-	-	-	-	-	2	1	1

CO2	2	2	-	-	-	-	-	-	-	-	2	1	1
CO3	3	3	-	-	-	-	-	-	-	-	2	1	2
CO4	3	3	-	-	-	-	-	-	-	-	2	1	2
CO5	3	3	-	-	-	-	-	-	-	-	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

ASSESSMENT SCHEME

Definitions

Continuous Internal Assessment (CIA)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) – Viva and Assignment (AI)	25
Mid Semester Examination (CBT)	15
Attendance	10
Total	**50 Marks**

RECOMMENDED TOOLS

Language Tools

- Google Translate
- Google Assistant
- OCR Applications

AI & NLP Tools

- ChatGPT
- NLP-based Translation Tools
- Voice Recognition Systems

Learning Platforms

- YouTube Kannada Learning Channels
- Online Language Learning Platforms
- Kannada Literature Digital Archives

SUGGESTED READING

PRIMARY TEXTBOOKS

1. Kannada Manasu – Engineering Kannada Text
2. Selected Kannada Literary Works
3. Works of Purandaradasa and Kuvempu
4. Kannada Sahitya Parishat Publications

ADDITIONAL RESOURCES

1. Kannada Literature Resources
2. Digital Kannada Learning Materials
3. AI-based Language Learning Platforms
4. Regional Language Preservation Resources

****

CALCULUS & MATHEMATICAL MODELING
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - II

Course Code	:		**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	30 Hrs L
L–T–P–J	:	3–0–0–0			

Course Vision:

To develop strong mathematical foundations in modelling, multivariable calculus, differential equations, and artificial intelligence concepts so that students can analyze complex engineering problems and design intelligent data-driven systems responsibly.

Course Objectives:

This course will enable students to:

1. Understand the principles of mathematical modelling for representing real-world systems.
2. Apply multivariable calculus concepts to analyze and optimize engineering and computational problems.
3. Evaluate multiple integrals and differential equations for modelling dynamic systems.
4. Apply Laplace transforms and analytical methods to solve differential equations arising in engineering applications.
5. Understand artificial intelligence search algorithms and reasoning methods used in intelligent decision-making systems

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture Method:** The course may be delivered through traditional lectures along with demonstrations using MATLAB/Python to explain Artificial Intelligence concepts, algorithms, and mathematical models.
2. **Interactive Teaching:** Encourage active learning through brainstorming, coding discussions, algorithm analysis, group problem-solving, and questioning sessions related to AI applications.
3. **Demonstration using MATLAB/Python:** Demonstrate AI algorithms such as classification, clustering, regression, and optimization using MATLAB or Python notebooks to help students visualize results and understand implementations.
4. **Multimedia Learning:** Show videos, simulations, and animations explaining machine learning models, neural networks, and real-world AI applications to improve conceptual understanding.
5. **Collaborative Learning:** Encourage students to work in groups to design small AI projects, implement algorithms, and analyze datasets using MATLAB or Python.
6. **Higher Order Thinking Questions:** Ask analytical and application-based questions that require students to design models, interpret AI outputs, compare algorithms, and evaluate model performance.

7. **Problem Based Learning:** Present real-world problems such as prediction, classification, pattern recognition, or optimization tasks where students must apply AI techniques using MATLAB/Python.

MODULE – I	**09 Hours**
Introduction to Mathematical Modelling	
Topics: Introduction to Mathematical Modelling: Definitions, importance of modelling in real-world problems, types of mathematical models, deterministic models, probabilistic models, static and dynamic models, functions and their role in modelling, function fitting and data modelling. **Applications:** Engineering system modelling, Population growth prediction, Economic forecasting, Climate modelling	
MODULE – II	**09 Hours**
Multivariable Calculus and Optimization	
Topics: Functions of two or more variables, level curves, limits and continuity, partial derivatives, chain rules, total differential, Jacobian, Taylor's expansion for two variables, linear approximation and increment estimation, maxima, minima and saddle points, constrained maxima and minima, Lagrange's multiplier method. **Applications:** Engineering design optimization, Economics and production optimization, Surface modelling in computer graphics, Robotics motion planning	
MODULE – III	**09 Hours**
Multiple Integrals	
Topics: Evaluation of double integrals, change of order of integration, change of variables between Cartesian and polar coordinates, evaluation of triple integrals, change of variables between Cartesian coordinates, error function and its properties. **Applications:** Volume and area computation in engineering systems, Probability distribution modelling, Fluid dynamics simulations, Heat distribution modelling	
MODULE – IV	**09 Hours**
Differential Equations and Transform Methods	
Topics: Introduction to first-order ordinary differential equations, linear higher order differential equations with constant coefficients, homogeneous and non-homogeneous equations, method of undetermined coefficients, method of variation of parameters, Laplace transforms and their applications. **Applications:** Control systems engineering, Electrical circuit analysis, Mechanical vibration systems, Population dynamics modelling	
MODULE – V	**09 Hours**
Artificial Intelligence and Reasoning	

Topics: AI search algorithms: informed and uninformed search, adversarial search, introduction to logic: propositional logic and predicate logic, reasoning under uncertainty, conditional independence representation, exact inference through variable elimination, approximate inference through sampling.

Applications: Game playing AI systems, Autonomous navigation systems, Expert systems for decision support, Intelligent recommendation systems

Authentic Intelligence:

1. Understanding limitations of models in real-world applications
2. Ethical interpretation of optimization results in AI systems
3. Correct interpretation of numerical results in data-driven systems
4. Understanding limitations of dynamic models
5. Ethical implications of automated decision making

AI Integration:

1. Mathematical models used in machine learning systems
2. Gradient computation in machine learning
3. Numerical computation in AI models
4. Dynamic system modelling in AI robotics
5. Search algorithms used in robotics and navigation.

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	To understand mathematical modelling concepts including deterministic, probabilistic, static, and dynamic models and apply function fitting techniques for real-world and AI data modelling.	L2
2	To apply multivariable calculus concepts including partial derivatives, Jacobians, Taylor expansion, and optimization methods such as Lagrange multipliers in AI and engineering problems.	L3
3	To evaluate double and triple integrals and apply coordinate transformations in computational modelling and probabilistic AI systems.	L4
4	To solve ordinary differential equations and apply Laplace transforms for modelling dynamic systems used in AI, signal processing, and control systems.	L3
5	Design and apply search algorithms, logical reasoning, and probabilistic inference techniques in artificial intelligence systems and intelligent decision-making models.	L6

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs													
	Program Outcomes (POs)											**PSOs**	
COS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO-1	3	3	2	2	2	1			1	1		2	–
CO-2	3	3	3	2	2				1	1		3	1
CO-3	3	3	2	3	2				1	1		3	3
CO-4	3	3	3	3	2	1			1	1		3	2
CO-5	3	3	3	3	3	1		1	2	2	1	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Thomas' Calculus, George B. Thomas, D. Weir and J. Hass, 2014, 13th edition, Pearson.
2. Erwin Kreyszig – Advanced Engineering Mathematics, Wiley.
3. Stuart Russell & Peter Norvig – Artificial Intelligence: A Modern Approach, Pearson.
4. Kenneth, K. R., Discrete Mathematics and its Applications, 7th Ed., Tata McGraw Hill

REFERENCE BOOKS:

1. Gilbert Strang – Linear Algebra and Learning from Data, Wellesley Cambridge Press.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, MIT Press.
3. Steven Chapra – Applied Numerical Methods with MATLAB, McGraw Hill.

E-RESOURCES:

RECOMMENDED TOOLS

- Python
- MATLAB
- NumPy
- Pandas
- Matplotlib
- Scikit-learn
- Jupyter Notebook

LAB EXERCISES

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Introduction to Mathematical Modelling (03 Hours)					
1	Curve Fitting Techniques.	Implement curve fitting techniques on real-world datasets.	MATLAB/PYTHON	1 h	**CO1**
2	Function Plotting & Analysis	Plot and analyze different types of mathematical functions	MATLAB/PYTHON	1 h	**CO1**

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
3	Polynomial Regression	Implement polynomial regression to approximate nonlinear data patterns.	MATLAB/PYTHON	1 h	**CO1**
MODULE II: Exercises 4–6 : Multivariable Calculus and Optimization (03 Hours)					
4	Tangent & Rate of Change	Visualize functions and their rates of change using tangent lines.	MATLAB/PYTHON	1 h	**CO2**
5	Single and multi-variable function	Evaluate maxima and minima (extrema) of a single and multi-variable function and Plot 3D surfaces	MATLAB/PYTHON	1 h	**CO2**
6	Lagrange Multipliers	Apply Lagrange Multiplier method	MATLAB/PYTHON	1 h	**CO2**
MODULE III: Exercises 7–9: Multiple Integrals (03 Hours)					
7	Integration & Area	Understand integration as area under the curve through visualization.	MATLAB/PYTHON	1 h	**CO3**
8	Volume Under Surfaces	Evaluate volume under surfaces using integration techniques.	MATLAB/PYTHON	1 h	**CO3**
9	Triple Integrals	Compute triple integrals for volume calculations	MATLAB/PYTHON	1 h	**CO3**
MODULE IV: Exercises 10–12: Differential Equations and Transform Methods (03 Hours)					
10	Differential Equations (Basic)	Solve first-order and higher-order differential equations computationally	MATLAB/PYTHON	1 h	**CO4**
11	Laplace Transform	Use Laplace transforms to solve initial value problems.	MATLAB/PYTHON	1 h	**CO4**
12	Case Study of Dynamic Modelling	Simulate time-series data using differential equation models.	MATLAB/PYTHON	1 h	**CO4**
MODULE V: Exercises 13–15: Artificial Intelligence and Reasoning (03 Hours)					
13	**Understanding the Search algorithm**	Implement informed search algorithm for pathfinding problems	MATLAB/PYTHON	1 h	**CO5**
14	**AI Techniques Implementation**	Implement search algorithms, logic inference, and Bayesian reasoning.	MATLAB/PYTHON	1 h	**CO5**
15	**Case Study of AI Search Algorithm**	Develop a simple propositional logic inference engine	MATLAB/PYTHON	1 h	**CO5**

Engineering Chemistry

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I /II

Subject Code :		**Credits**	: 03
Hours / Week :	03 Hours	**Total Hours**	: 45 Hrs (30Hrs L +15Hrs P)
L–T–P-J :	2–0–2-0		

Course Vision:

To impart strong fundamentals in chemical sciences and advanced materials, integrating computational tools, modern electronic systems, and sustainable energy technologies, enabling students to innovate and contribute effectively to contemporary engineering solutions.

Course Learning Objectives:

This Course will enable students to:

1. Understand atomic structure, quantum principles, and chemical bonding theories to explain molecular properties.
2. Understand computational tools for molecular modeling, structure building, and basic property prediction.
3. Understand principles and applications of sensors, electrochemical systems, and modern energy devices.
4. Understand polymer properties and explore green energy technologies including photovoltaics and hydrogen fuel.
5. Understand electronic memory systems, optoelectronic materials, and their advanced applications.

BROAD COURSE DESCRIPTION

This course provides a comprehensive foundation in engineering chemistry by integrating fundamental concepts of atomic structure, chemical bonding, and molecular properties with modern applications. It introduces students to polymers, electrochemical systems, sensors, and advanced materials, including optoelectronic and electronic memory devices. The course also emphasizes sustainable technologies such as green fuels, photovoltaics, and hydrogen energy, as well as computational tools for molecular modeling, enabling students to apply chemical principles to real-world engineering problems.

UNIT – I	09 Hours

QUANTUM AND MOLECULAR FOUNDATIONS FOR COMPUTING MATERIALS

Atomic Structure -de Broglie's theory, Bohr's atomic model-Sommerfeld's extension of atomic structure; Electronic configuration and Quantum numbers; Shapes of s, p, d, f orbitals - Pauli's exclusion principle - Hund's Rule of maximum multiplicity- Aufbau principle. Heisenberg's uncertainty principle – wave nature of electron – Schrodinger wave equation (No derivation). Chemical bonding and hybridization of atomic orbitals involving s, p and d orbitals.

Solutions: Types of solutions and classification (solid, liquid and gaseous solutions). Concentration units—molarity, molality, normality, mole fraction and mass percentage. Preparation of solutions and numerical problems involving molarity. Solubility and factors affecting solubility.

Engineering Applications: Semiconductors and electronic devices, prediction of molecular behavior and material properties, Development of nanotechnology and functional materials, Optimization of chemical and industrial processes, Environmental engineering, Corrosion control.

UNIT – II	07 Hours

MOLECULAR MODELLING AND CHEMINFORMATICS

Existence and uniqueness of solutions for ordinary differential equations. Solving initial value problems using Taylor series method, Euler's method and modified Euler method, Runge-Kutta methods, multistep methods. Higher-order equations and systems. Stability analysis. Solving boundary value problems using shooting methods, difference methods and variational methods. Introduction to numerical solutions for partial differential equations.

Engineering Applications: Weather prediction models, Mechanical system simulations, Control system modelling, Heat transfer and diffusion problems.

UNIT – III	10 Hours

SENSOR TECHNOLOGIES AND ELECTROCHEMICAL ENERGY SYSTEMS Topics: Linear transformations, Matrices of linear transformations, Vector space of linear transformations, Inner product spaces, Orthogonal vectors, Projections, Gram-Schmidt orthogonalization process, Eigenvalues and eigenvectors and interpretation.

Applications: Image processing, Computer vision systems, Signal filtering, Structural engineering analysis.

UNIT – IV	09 Hours

POLYMERIC MATERIALS AND SUSTAINABLE ENERGY TECHNOLOGIES Polymers: Introduction, Molecular weight - Number average, weight average, and numerical problems. Conducting polymers – Preparation, properties, and commercial applications of conducting polymers. PANI (polyaniline), PPy (polypyrrole), and PEDOT (poly(3,4-ethylene dioxythiophene). Specialty of Polymers for designing computers: Preparation, Properties and Applications. Acrylonitrile-butadiene-styrene (ABS) and polycarbonate (PC), and PVC. Green Fuels: Introduction to different types of fuels, and perspective of green fuels (Bioethanol, and Biodiesel). Photovoltaic cells: Introduction, definition, importance, working of PV cell. Solar grade silicon physical and chemical properties relevant to photo-voltaic, doping of silicon by diffusion technique. Green hydrogen: Introduction to properties of hydrogen pertaining to fuel. Introduction to electrolysis of water. Generation of hydrogen by electrolysis of water and its advantages.

Engineering Applications: Flexible electronics, design of plastic components in computers, clean energy and sustainable energy.

UNIT – V	09 Hours

ELECTRONIC MEMORY AND OPTOELECTRONIC MATERIALS

Topics: Memory Devices: Introduction, Basic concepts of electronic memory, History of organic/polymer electronic memory devices. Classification of electronic memory devices (Transistor-Type, Capacitor-Type, Resistor-Type and Charge transfer type Electronic Memory devices), types of organic memory devices. Display Systems: Photoactive and electroactive materials - Definition and principle. Optoelectronic devices: Definition, working principle. Nanomaterials (Silicon Nanocrystals) and organic materials [Light absorbing materials - Polythiophenes (P3HT), Light emitting materials – Poly [9-vinylcarbazole] (PVK)]. Four properties of optoelectronic devices.

Engineering Applications: Memory devices, OLEDs, sensors, nanoelectronics.

LAB EXERCISES

The lab emphasizes **hands-on Experience and visualization**.

Lab Experiments

Volumetric and Instrumental Analysis

1. Potentiometric estimation of FAS using standard potassium dichromate.
2. Conductometric estimation of H^+ in a weak and strong acid using standard sodium hydroxide solution.
3. Determine pKa of a weak acid using standard sodium hydroxide with a pH meter.
4. Synthesis of Nylon 6,10 (Condensation Polymer).
5. Colorimetric estimation of copper in a given solution.

6. Electrolysis of water.

Computational experiments

7. Computational Investigation of Ethylene Carbonate-Li+ Complexes: Structural and Vibrational Properties in Lithium-Ion Batteries.
8. Molecular Dynamics Simulation and Structural Analysis of Solvated Lysozyme.
9. Computational IR Spectroscopy of Pharmaceuticals: Analyzing Paracetamol and Phenacetin.
10. Computational Studies of Chain Length Effects on the Optoelectronic Properties of PEDOT.
11. Computational Investigation of the Pharmacophore Quinoline: Electronic and Thermodynamic Profiling.
12. Computational Conformational Analysis of Substituted Cyclohexanes.
13. Cheminformatics and ADME Profiling of Drug Candidates.

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration

Module-1: Students explore and visualize atomic and molecular structures using interactive platforms like Atom Animation and MolView, enabling them to understand spatial arrangement, bonding, and properties of molecules in solutions through dynamic 2D/3D models, thereby connecting theoretical chemistry with real-time digital learning and AI-assisted visualization.

Module-2: Conceptual overview of AI and Machine Learning models interpreting SMILES for property prediction and drug discovery (Python for chemistry using Google Colab and RDKit for programmatic handling of molecules and automated data extraction).

Module-3: Collect, visualize, and analyze sensor data using MATLAB and AI tools for real-time IoT-based projects and predictive analytics. https://thingspeak.mathworks.com/

Module-4: Explore, analyze, and predict materials for energy systems such as batteries, and fuel cells using computational and AI-driven approaches. (https://next-gen.materialsproject.org/)

Module-5: AI-based prediction of polymer properties (Tg, conductivity, strength), material selection using data-driven models, simulation of polymer behavior, AI-assisted design of smart and flexible materials.

Authentic Intelligence Reflection:

Module-1: Students connect theory with real-world chemistry, enhancing spatial understanding and critical thinking. AI-assisted exploration fosters analysis of structure–property relationships, promoting problem-solving skills aligned with modern computational and research-driven scientific practices.

Module-2: This module builds authentic intelligence by enabling students to apply AI and machine learning for interpreting SMILES in property prediction and drug discovery. Using Python, Google Colab, and RDKit, learners gain hands-on experience in molecular data handling, fostering computational thinking, automation skills, and real-world problem-solving in modern cheminformatics and pharmaceutical research.

Module-3: This module fosters authentic intelligence by engaging students in real-time IoT data collection, visualization, and analysis using MATLAB and AI tools. By working with platforms like ThingSpeak, learners develop data-driven decision-making and predictive analytics skills, connecting theoretical knowledge with practical applications in smart systems, automation, and modern engineering problem-solving environments.

Module-4: This module develops authentic intelligence by enabling students to explore and predict materials for energy systems using computational and AI-driven approaches. Using platforms such as the Materials Project, learners analyze structure–property relationships, fostering data-driven reasoning, critical evaluation, and innovation skills aligned with modern materials discovery and sustainable energy research.

Module-5: This module cultivates authentic intelligence by integrating AI-driven prediction of polymer properties such as Tg, conductivity, and strength. Students apply data-driven models for material selection and simulate polymer behavior, fostering analytical thinking. AI-assisted design of smart, flexible materials connects theory with innovation, aligning learning with contemporary computational materials science and engineering practices.

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Apply the principles of atomic structure, quantum mechanics, chemical bonding, and solution chemistry to explain molecular properties, predict structures, and solve concentration and solubility-related problems.	L3
CO2	Apply computational chemistry tools to model molecular structures, and construct Z-matrices, evaluate chemical properties.	L4
CO3	Apply principles of sensors, electrochemical systems, and energy devices to analyze sensing mechanisms, battery technologies, and quantum dot solar cell applications.	L3
CO4	Apply the concept of polymerization for the synthesis of polymers and study their structure-property relationship for commercial applications.	L3
CO5	Apply concepts of electronic memory and optoelectronic materials to understand device functioning and material applications.	L5

MAPPING LEVELS OF COS TO POS / PSOS

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO-1	3	3	-	-	-	-	-	-	-	-	2	1	1
CO-2	3	3	-	-	-	-	-	-	-	-	2	1	2
CO-3	3	3	-	-	-	-	-	-	-	-	2	1	2
CO-4	3	2	-	-	-	-	-	-	-	-	2	1	1
CO-5	3	3	-	-	-	-	-	-	-	-	2	1	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

RECOMMENDED TOOLS

TEXT BOOKS:

1. Howard Anton & Chris Rorres, *Elementary Linear Algebra with Applications*, Wiley.
2. Gilbert Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press.
3. Sheldon Ross, *A First Course in Probability*, Pearson.
4. Douglas C. Montgomery & George C. Runger, *Applied Statistics and Probability for Engineers*, Wiley.

REFERENCE BOOKS:

Engineering Chemistry – I: Concepts and Applications, Jit Chakraborty, Wiley India Pvt. Ltd.

2. *Engineering Chemistry*, Shikha Agarwal, Cambridge University Press.
3. *Molecular Modelling for Beginners*, Alan Hinchliffe, Wiley, 2008.
4. *A Textbook of Engineering Chemistry*, R. Gopalan, D. Venkappayya & Sulochana Nagarajan, S. Chand Publishing.
5. *Engineering Chemistry*, Jain & Jain, Dhanpat Rai & Co.
6. *Materials Science and Engineering: An Introduction*, William D. Callister Jr., Wiley India Pvt. Ltd.

Additional Resources

1. NPTEL: Database Management Systems – IIT Madras / IIT Kharagpur
2. Coursera: Database Systems Specialization by University of Michigan
3. NPTEL: Engineering Chemistry – IIT Madras
4. NPTEL: Electrochemistry / Sensors – IIT Kharagpur
5. NDLI: National Digital Library of India – IIT Kharagpur
6. LibGen: Online Academic Resource Repository
7. YouTube: Engineering Chemistry Lectures – NPTEL / IIT Channels
8. YouTube: Electrochemistry and Sensors – Educational Lectures
9. YouTube: Polymers and Materials Science – Concept Videos
10. YouTube: Photovoltaic Cells and Green Energy – Lecture Series
11. YouTube: Optoelectronic Devices and Nanomaterials – Tutorials

Reference book

1. Dayananda Sagar University laboratory manual.
2. J. Bassett, R.C. Denny, G.H. Jeffery, Vogels, Text book of quantitative inorganic analysis, 4th Edition.
3. Prasanta Rath, “Engineering Chemistry”, Cengage Learning India PVT, LTD, Delhi, 2015.
4. Shikha Agarwal, “Engineering Chemistry-Fundamentals and Applications”, Cambridge University Press, Delhi, 2015.

****

Engineering Graphics & Generative Design

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 03
Hours / Week :	03 Hours	**Total Hours**	: 45 Hrs (30Hrs L +15Hrs P)
L–T–P–J :	2–0–2-0		

Course Vision: This course introduces engineering graphics, parametric modeling, and generative design integrated with computational thinking and automation. It enables CSE students to bridge design and programming through CAD tools, AI-assisted workflows, and Fusion API-based automation.

Prerequisites:

Course Learning Objectives:

Students will be able to:

6. To develop fundamental understanding of engineering graphics and visualization
7. To enable creation of parametric 2D and 3D CAD models
8. To introduce AI-driven generative design concepts
9. To apply programming concepts for CAD automation using Fusion API
10. To build computational design and automation skills

BROAD COURSE DESCRIPTION

This course introduces the fundamentals of engineering graphics, parametric modeling, and generative design with a strong emphasis on computational thinking and automation. It is specifically designed for CSE cluster students to bridge the gap between design and programming by integrating CAD tools, AI-assisted workflows, and API-based development.

Students begin with engineering drawing principles and 2D sketching, progressing to parametric 3D modelling and assemblies. The course then explores generative design, where AI-driven techniques are used to create optimized and innovative design solutions based on constraints and objectives. Further, students learn to program and automate CAD tasks using Fusion API and Python, enabling the development of intelligent design systems.

By the end of the course, students will be able to create, analyze, and automate design processes, develop data-driven and parametric models, and apply computational approaches to real-world engineering design problems. The course emphasizes both AI integration and authentic intelligence, ensuring students not only use intelligent tools but also critically interpret and validate their outputs.

UNIT – I	08 Hours

Engineering Graphics & 2D Sketching: Engineering graphics fundamentals, BIS standards, drawing conventions, Orthographic projections, sectional views, isometric views, CAD interface and sketch environment, 2D sketching: tools, constraints, dimensioning

Engineering Applications: Technical drawing and documentation

UNIT – II	06 Hours

3D Modeling: Parametric and feature-based modelling, Solid modeling: extrude, revolve, sweep, loft, Feature modifications and assemblies

Engineering Applications: Product design and digital prototyping

UNIT – III	09 Hours

Generative Design: Generative design concepts and workflows, Constraints, materials, loads, Optimization and validation.

Engineering Applications: Lightweight and optimized structures

UNIT – IV	14 Hours
Fusion API & CAD Programming:Introduction to APIs and Fusion API architecture, Python scripting fundamentals for CAD, Accessing and modifying sketches, features, and parameters, Automating 2D/3D model creation using scripts, Event handling and UI basics in Fusion, Debugging and execution workflows **Engineering Applications:** Automated CAD model generation, parametric script-based design, Custom CAD tool development	
UNIT – V	**08 Hours**
Design Automation & Computational Design: Design automation workflows and pipelines, Parametric and rule-based design systems, Generating design variants using code, Data-driven design (inputs → geometry), Introduction to add-in development, CAD + data + automation integration **Engineering Applications:** Mass customization systems, automated design exploration tools, Smart CAD applications	

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

The course follows an **|A|^2 Pedagogy** (AI Integration & Authentic Intelligence Reflection).

- **AIM:** To bridge the gap between design and programming through CAD tools, AI-assisted workflows, and API-based development. Students must understand foundational logic where AI-generated outputs are critically evaluated against engineering principles and computational logic.
- **AI Logic Mirroring:** The pedagogy emphasizes validating AI outputs using design intent, constraints, and feasibility analysis, mirroring the decision-making process of neural networks (input → logic → output validation).
- **Example Implementation (Perceptron Logic):** While the provided PDF focuses on CAD and Python for Fusion API, it emphasizes **Algorithmic Thinking** and **Problem Decomposition**. Students are expected to:
  - o Develop weighted sum functions and activation logic within the context of **Design Automation**.
  - o Use AI-assisted code generation for CAD scripting and perform AI-based debugging and error detection.

COURSE OUTCOMES (CO)

CO	Course Outcome	Bloom's Level
CO1	Apply engineering graphics principles and create 2D sketches	L3
CO2	Develop parametric 3D models and assemblies	L3
CO3	Analyze and interpret generative design outputs	L4
CO4	Implement CAD automation using Python and Fusion API	L3
CO5	Design computational workflows for automated design systems	L3

ASSESSMENT SCHEME

Definitions

1. **Internal Assessment (CIA) – 60 Marks**

a. One **Mid Semester Examination (MSE)** (CAD Tool-based, Laboratory Exercises),

b. One **Continuous Assessment Tool (CAT)**: It may include: Surprise test (Computer-based), Quiz, Project based learning, Reflection Note on Guest Lecture, Industrial visit, E-course certification, Building models, Group discussion, Case study, Seminar, or Open Book Assignment.

c. One **AI-powered Assignment (APA):** Use of AI tools for problem-solving while simultaneously developing Authentic Intelligence through critical evaluation, ethical reasoning, and responsible decision-making.

OR

d. **Mini Project:** Lightweight generative structural component design.

2. **"External Assessment"** refers to the Semester End Examinations (SEE). The weightage of this component is **40%**. At the end of semester, the final examination is conducted to evaluate student's performance.

1. Internal Assessment (CIA) – 60 Marks

Component	**Marks**	**Modules**
One MSE	30	Module 1 & 2
CAT - Assignment on Generative Design	10	Module 3
AI-powered Assignment (APA) **OR** Mini Project	20	All Modules
Total	**60**	—

2. External Assessment (SEE) – 40 Marks

Component	**Marks**	**Module**
Exercise 1 Using CAD Tool	20	Module 4
Exercise 2 Using CAD Tool	20	Module 5
Total	**40**	

MAPPING LEVELS OF COS TO POS / PSOS

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO-1	3	2	1		1					2		2	1
CO-2	3	3	2		1					1		2	1
CO-3	2	1	2		3				1	2		3	2
CO-4	2	2	3	2	3	1	2		1	1		3	3
CO-5	1		1		2			1	2	3	1	1	1

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Recommended Tools

- CAD Software: Autodesk Fusion
- Simulation: Generative design tools, CAD simulation modules
- Programming: Python (Fusion API)
- AI Tools: ChatGPT, CAD plugins

Suggested Reading

Textbooks

- K. Venugopal, *Engineering Drawing and Graphics*, New Age International
- Sham Tickoo, *Autodesk Fusion 360 for Designers*, CADCIM Technologies

Reference Books

- Ibrahim Zeid, *CAD/CAM: Theory and Practice*, McGraw Hill
- Ian Stroud, *3D Modelling and CAD*, Springer

Additional Learning Resources

- NPTEL – Engineering Graphics and CAD
- Autodesk Learning Resources (Fusion 360 official tutorials)
- Coursera / edX / Udemy MOOCs on CAD, Generative Design, and Computational Design
- Research papers from IEEE Xplore / ScienceDirect (for advanced exploration)

Mapped NVIDIA DLI Course

- Course Title: Fundamentals of Deep Learning OR Computer-Aided Engineering (Simulation & Digital Twins)
- Focus: Understanding how neural networks and parallel processing (CUDA) accelerate design validation and generative iterations.
- Relevance: This course provides the foundational knowledge needed to understand the "AI Logic" (Input → Weighted Sum → Activation) mentioned in your syllabus's AI Integration section.

DLI Learning Path for Engineering Design

This path is designed to transition students from traditional CAD to AI-accelerated generative workflows**Laboratory Component (1 Credit)**
The laboratory component focuses on Autodesk Fusion 360 and Python-based automation.

Lab Experiments (Indicative):

2D Sketching: Creating sketches with auto-dimensioning and sketch recognition tools.
3D Modeling: Developing parametric 3D models using extrude, revolve, and assembly constraints.
Generative Design: Applying AI-driven exploration based on materials, loads, and constraints.
Fusion API Scripting: Writing Python scripts to access and modify CAD parameters and automate model creation.
Design Automation: Building rule-based design systems and parametric product configurators.
Advanced Automation: Using AI-assisted code generation and debugging to optimize Python-based CAD scripts.

AI-Integrated Lab Exercises (Indicative)

· Exercise 1 (Visualization): Use AI-based visualization for orthographic, sectional, and isometric views.
· Exercise 2 (Optimization): Use cloud-based generative design tools for performance prediction and structural optimization.
· Exercise 3 (Automation): Implement AI-assisted code generation for CAD scripting and automate parametric modeling workflows.
· Exercise 4 (Validation): Critically evaluate AI-generated designs against engineering standards (e.g., BIS standards) and manufacturability.
The course emphasizes that students must debug AI-generated scripts, validate automation logic, and analyze trade-offs between performance and complexity.

****

Human-Centered Design, Ethics & Values

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 02
Hours / Week :	02 Hours	**Total Hours**	: 30 Hrs (15Hrs L +15Hrs T)
L–T–P-J :	1–1–0-0		

Course Vision: To enable students to develop empathy, design user-centric solutions, and understand the ethical implications of technology on society

Prerequisite:

Course Learning Objectives:

This Course will enable students to:

BROAD COURSE DESCRIPTION

This course introduces computational thinking and problem-solving using Python programming. It covers fundamental concepts such as control structures, functions, strings, data structures, and object-oriented programming with an emphasis on modular and efficient coding. The course incorporates the use of standard Python libraries for data handling and processing, along with file handling and exception handling to develop reliable programs. Students gain practical experience through laboratory exercises focused on real-world problem solving. The course also introduces foundational artificial intelligence concepts such as data preprocessing and basic pattern identification using Python libraries, along with awareness of modern computing paradigms such as accelerated computing for handling large-scale data applications. The course prepares students to design and implement structured Python-based solutions.

UNIT – I	06 Hours

Foundations of Human-Centered Design (HCD)

Topics: Introduction to HCD vs. Product-Centered Design; The Design Thinking Process (Empathize, Define, Ideate, Prototype, Test); Understanding user experience (UX) and human limitations (human error as a design failure); Cognitive ergonomics and mental models.

Activity: Identifying a "bad design" in daily life and mapping the user journey.

UNIT – II	06 Hours

Empathy, Research, and User Understanding

Topics: Research methods: Observation, User Interviews, Focus Groups; Empathy mapping; Creating User Personas; Contextual inquiry; Understanding diverse societal contexts.

Activity: Conducting a 10-minute user interview and empathy map creation.

UNIT – III	07 Hours

Ethical Frameworks and Values in Engineering

Topics: Topics: Engineering Ethics: Morals, Values, and Ethics; Moral dilemmas, dilemmas in technological design; Introduction to ethical theories (Utilitarianism, Duty Ethics, Virtue Ethics); Professionalism and codes of ethics (IEEE/ACM).

Activity: Case study discussions on ethical dilemmas (e.g., Tesla autopilot, social media algorithms).

UNIT – IV	05 Hours

Human-Centered Design for Sustainability and Safety

Topics: Designing for Safety: Risk-benefit analysis and reducing risk; Environmental Ethics: Sustainable

Design Transitions; Social Responsibility; Accessibility and Inclusive Design (Designing for people with limitations). **Activity:** Redesigning a product interface for accessibility or sustainable materials.	
UNIT – V	**06 Hours**
Responsible Innovation and Societal Impact **Topics:** AI Ethics and Data Privacy; Algorithmic bias and accountability; Intellectual Property Rights (IPR) and Plagiarism; Future of humanity and technology; Ethical AI and autonomous systems. **Activity:** Designing an ethical technology solution (e.g., an inclusive app or safe gadget).	

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Differentiate between user-centric and technology-centric approaches.	L2
CO2	Ability to conduct research to understand user needs and emotional context.	L2
CO3	Apply moral reasoning to identify ethical issues in engineering scenarios.	L3
CO4	Integrate safety and environmental stewardship into design solutions.	L3
CO5	Evaluate the long-term societal impact of technology and design responsibly.	L4

STRENGTH OF CO MAPPING TO PO/PSOS:

Table: Mapping Levels of Cos to Pos / PSOs

COs	Program Outcomes (Pos)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	2	2									2	1	1
CO2	2	2									2	1	1
CO3	3	3									3	1	2
CO4	3	3									3	1	2
CO5	3	3									3	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

AI Integration:

- Design Thinking Support: Use AI-powered tools (e.g., empathy mapping software, sentiment analysis) to analyze user interviews and generate personas.
- Ethics Case Simulations: AI-driven scenario generators can present ethical dilemmas (e.g., autonomous

vehicles, algorithmic bias) for students to evaluate.
- Accessibility & Inclusivity: AI-based design platforms can help prototype accessible interfaces (voice-to-text, screen readers, adaptive layouts).
- Data Privacy Awareness: Introduce students to AI auditing tools that detect bias, privacy risks, and compliance issues in design solutions.
- Collaborative Ideation: AI brainstorming assistants can support ideation sessions, offering diverse design alternatives while students critically evaluate feasibility.

Authentic Intelligence Reflection:

- **Critical Thinking:** Students reflect on how AI suggestions align—or conflict—with human empathy, values, and ethical frameworks.
- **Human Judgment:** Emphasize that while AI can generate options, authentic intelligence (human reasoning) is required to choose ethically sound solutions.
- **Societal Responsibility:** Encourage reflection on long-term impacts of AI-driven designs on sustainability, inclusivity, and cultural contexts.
- **Ethical Balance:** Students assess when to rely on AI tools and when to prioritize human-centered decision-making.
- **Personal Growth:** Journaling or group discussions on how AI influenced their design process, and whether it enhanced or limited their authentic creativity.

LABORATORY COMPONENT

The laboratory emphasizes hands-on implementation of Python programs for problem solving. Students will use IDLE, VS Code, Jupyter Notebook or Google Colab and standard libraries (such as *math, random, datetime, re, json, logging, NumPy, Pandas,* and *Matplotlib*) to develop, analyze, and validate solutions for data processing and computational tasks.

Lab Experiments

1. Develop a program for a temperature alert system that triggers warnings when temperature exceeds a threshold using the random module for simulated input.
2. Develop a traffic signal control system using conditional statements and loops.
3. Create a log file analyzer to detect errors and generate reports using re for pattern matching.
4. Build a text validation system for healthcare or retail data inputs using re for format validation.
5. Develop a program to perform file operations for storing and retrieving records using json for structured data storage.
6. Develop a shopping cart application using lists and dictionaries to add, remove, update items, and calculate total bill with discounts.
7. Develop a student record management system using lists and dictionaries with file handling to add, update, delete, and retrieve student records, and store data persistently.
8. Develop a program to analyze sales data using Pandas, compute total sales, identify top-selling products, and visualize trends using Matplotlib.
9. Perform CSV data analysis and visualization using Pandas and Matplotlib libraries to load data, clean and process records, compute summary statistics, and generate visual representations such as bar charts or line graphs.
10. Develop an object-oriented banking system to create and manage accounts, perform operations such as deposit, withdrawal, and balance inquiry, and implement transaction handling using classes and methods.
11. Implement matrix operations using operator overloading with optional use of NumPy for comparison.
12. Develop a robust input validation system with exception handling and use of logging for error tracking.
13. Develop an ATM system using functions and file handling to perform operations such as balance inquiry, deposit, and withdrawal with proper validation and exception handling.

AI-Integrated Lab Exercises (Indicative)

5. Develop a Python program to load a dataset (CSV file) using Pandas, perform preprocessing (handling

missing values, sorting, filtering), and compute basic statistics (mean, max, min). Visualize the results using Matplotlib.

6. Develop a Python program using math and random libraries to: Generate random numbers, Perform mathematical operations (square root, logarithm, trigonometric functions), Simulate a simple scenario (e.g., random score generation and analysis)
7. Develop a Python program to load a dataset (e.g., student marks) using Pandas and classify records into categories such as *pass/fail* or *high/medium/low* based on predefined conditions. Visualize the classification results using Matplotlib.
8. Develop a Python program to use the datetime library to process timestamps and analyze time-based data (e.g., login records or transactions). Store and process the data using Pandas, and generate summary insights.

REFERENCES

1. Don Norman, "Design of Everyday Things", MIT Press, Revised and Expanded Edition, 2013.
2. Mike W. Martin and Roland Schinzinger, "Ethics in Engineering", 4th Edition, McGraw-Hill Education, 2005.
   https://shaider.weebly.com/uploads/5/7/2/6/57266267/ethics_in_engineering_2nd_edition.pdf or
   https://studentebookhub.com/wp-content/uploads/2024/preview/9781260721744.pdf
3. R.S. Naagarazan, "Human Values and Professional Ethics", 3rd Edition, New Age International Publishers, 2022.
4. E. Balagurusamy, "Design Thinking: A Beginner's Perspective" 1st Edition, McGraw Hill Education, 2024.

****

DESIGN STUDIO–I: ENGINEERING PROBLEM SOLVING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 02
Hours / Week :	03 Hours	**Total Hours**	: 15 Hrs L + 15 Hrs J
L–T–P-J :	1–0–0-2		

Course Vision: Design Studio–I is a foundational experiential learning course designed to transform first-year engineering students from passive learners into active problem solvers, system thinkers, and prototype builders. The course introduces students to:

- Engineering problem-solving,
- Human-centered design,
- Sustainability-driven thinking,
- AI-assisted exploration,
- Teamwork,
- Prototyping, and
- Authentic Intelligence (AI²).

Students work in interdisciplinary teams to identify, analyze, design, prototype, and present solutions to real-world engineering and societal problems

Course Learning Objectives:

The course enables students to:

1. Understand real-world engineering problems through observation and analysis.
2. Apply structured problem-solving approaches.
3. Develop creativity, teamwork, and systems thinking.
4. Build low-fidelity and functional prototypes.
5. Use AI tools responsibly for ideation and analysis.
6. Understand sustainability, ethics, and societal impact.
7. Communicate engineering ideas effectively.

BROAD COURSE DESCRIPTION

This course introduces students to the foundations of **engineering problem-solving** through a structured design studio approach. It emphasizes identifying and framing problems in real-world contexts, generating multiple solution alternatives, and developing prototypes to validate ideas. Students will learn to integrate **computational thinking, design exploration, and responsible use of AI tools** into their workflow.

The course blends **analytical reasoning, creativity, and ethical reflection**, preparing students to address engineering challenges with sustainability, inclusivity, and societal impact in mind. Through hands-on activities, collaborative projects, and reflective exercises, learners will gain practical experience in applying engineering principles to solve complex problems while considering user needs, constraints, and broader contexts.

UNIT – I	09 Hours

UNDERSTANDING ENGINEERING PROBLEMS

Topics: What is engineering problem solving?, Problem identification & problem framing, Stakeholder analysis, Observation & empathy mapping, Real-world engineering challenges

Studio Activities

- Campus/community observation
- Problem identification workshop
- Team formation

AI Integration: AI-assisted idea exploration and AI-based brainstorming tools

Authentic Intelligence Focus: Difference between solving symptoms vs root causes and Human-centered thinking.

Deliverable: Problem Definition Report .

UNIT – II	09 Hours

DESIGN THINKING & IDEA GENERATION

Topics: Design thinking fundamentals, Ideation techniques, Functional decomposition, Systems thinking basics, Engineering constraints.

Studio Activities

- Brainstorming sessions
- Concept sketching
- Decision matrix

AI Integration: AI-generated alternatives and Comparing AI vs human ideas.

Authentic Intelligence Focus: Evaluating practicality & ethics and Trade-off analysis.

Deliverable: Concept Design Portfolio

UNIT – III	09 Hours

MODELING, SIMULATION & PROTOTYPING

Topics: Rapid prototyping, Basic modeling tools, Sensors & simple automation, Introduction to simulations, Low-fidelity & functional prototypes.

Studio Activities

- Prototype development
- Makerspace activities
- Basic simulations / Mockups

AI Integration: AI-assisted coding / design suggestions and AI for visualization.

Authentic Intelligence Focus: Reliability & usability and Failure analysis.

Deliverable: Prototype Version–1.

UNIT – IV	09 Hours

TESTING, ITERATION & VALIDATION

Topics: Testing methodologies, User feedback, Performance metrics, Iterative improvement, Engineering optimization basics.

Studio Activities

- Prototype testing
- Peer reviews
- Design iteration

AI Integration: AI-based analysis & visualization and AI-assisted debugging.

Authentic Intelligence Focus: Biases in AI suggestions and Responsible evaluation.

Deliverable: Prototype Version–2 and Testing Report

UNIT – V	09 Hours

SUSTAINABILITY, ETHICS & IMPACT

Topics: Sustainability in engineering, Lifecycle thinking, Ethical engineering practices, Inclusive technologies, Social impact assessment

Studio Activities

- Sustainability analysis
- Ethical dilemma discussions

AI Integration: AI and sustainability and Environmental impact estimation

Authentic Intelligence Focus: "Can this be built?" vs "Should this be built?"

Deliverable: AI² Reflection & Sustainability Report

UNIT – VI	09 Hours

FINAL INTEGRATION & DEMONSTRATION

Topics: Technical storytelling, Engineering presentations, Demo preparation, Innovation pitching.

Studio Activities

- Final prototype demonstration
- Poster presentation
- Jury evaluation

Deliverable: Final Prototype or Demo Video, Presentation and Technical Report

COURSE OUTCOME:

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course, the student will be able to:		
1	Identify and frame engineering problems considering users, constraints, and context	L2
2	Generate multiple design alternatives using engineering and computational thinking	L3
3	Develop basic prototypes/models to demonstrate solution concepts	L3
4	Use AI tools for ideation, analysis, and design exploration responsibly	L3
5	Analyze sustainability, ethics, and societal impact of engineering solutions	L4

Mapping Levels of COs to POs / PSOs

Os				Program Outcomes (POs)								SOs	
		1	2	3	4	5	6	7	8	9	10	11	**o1**	**SO 2**
O1		3	3									2	1	
O2		3	3									2	1	
O3		3	2									2	1	
O4		3	2									3	1	

O5		3	3									3	1	
3: Substantial (High)					**2: Moderate (Medium)**						**1: Poor (Low)**			

SAMPLE PROBLEM STATEMENTS

Students may work on:

- Reducing water and energy wastage in hostels
- Smart waste segregation
- Air quality monitoring
- Smart parking system
- Campus accessibility solutions
- Automated irrigation systems
- Pedestrian safety systems
- Smart classroom utilization

CoE & Infrastructure Utilization

Students may utilize:

- AI Makerspace,
- General Makerspace,
- AI & Robotics CoE,
- AR/VR/XR Lab,
- AI Factory (introductory exposure), or
- Any suitable lab.

ASSESSMENT SCHEME

Component	Weightage
Continuous Studio Evaluation	30%
Design Reviews	15%
Prototype Development	20%
AI² Reflection	10%
Final Demonstration & Viva	15%
Documentation & Presentation	10%

PROGRAM OUTCOMES MAPPING

PO	Mapping
PO1	Engineering knowledge
PO2	Problem analysis
PO3	Design/development of solutions
PO5	Modern tool usage
PO6	Engineer & society
PO7	Sustainability

PO	Mapping
PO9	Individual & teamwork
PO10	Communication
PO12	Lifelong learning

ASSESSMENT SCHEME :

Internal (100 Marks – No End Semester Exam Recommended)

Component	Marks
Continuous Assessment (Activities)	30
Design Challenge / Assignments	20
Prototype Development	20
IP Report / Patent Search	10
Final Project + Presentation	20

Recommended Tools

- **Design Tools:** Figma, Canva, Adobe XD
- **Prototyping:** Balsamiq, Tinkercad
- **AI Tools:** ChatGPT, Ideation tools, UI generators
- **IP Tools:** Google Patents, WIPO databases

Integrated Tools for Studio

- **Software:** Figma, Canva, Tinkercad
- **Platforms:** Google Forms (user feedback), Miro (brainstorming)
- **AI Tools:** Generative AI tools for ideation and prototyping
- **IP Platforms:** Google Patents, WIPO

TEXT BOOKS:

1. Change by Design – Tim Brown
   Covers fundamentals of design thinking, innovation strategies, and human-centered design (CO1, CO2, CO3).
2. Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days – Jake KnappFocuses on rapid prototyping, ideation, and solution validation
3. WIPO Intellectual Property Handbook – World Intellectual Property Organization
   Provides comprehensive understanding of IPR concepts, laws, and protection mechanisms (CO1, CO5).

REFERENCE BOOKS:

1. IDEO Design Thinking Resources
   Practical toolkits, case studies, and real-world applications of design thinking (CO2, CO3, CO4).
2. Stanford school Materials
   Structured methodologies, design frameworks, and collaborative innovation practices

3. The Design of Everyday Things – Don Norman
   Enhances understanding of user experience and usability principles (CO2, CO3).
4. Creative Confidence – Tom Kelley, David Kelley

E-RESOURCES : (RECOMMENDED TOOLS)

1. World Intellectual Property Organization – Official Learning Resources
https://www.wipo.int
(IPR tutorials, case studies, global standards
2. IDEO U – Design Thinking Courses
https://www.ideou.com
(Hands-on innovation and design thinking modules
3. Stanford University – d.school Open Resources
https://dschool.stanford.edu/resources
(Design thinking toolkits and guides
4. Coursera / edX

****

Engineering Science Course II- Introduction to C Programming
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 03
Hours / Week :	03 Hours	**Total Hours**	: 30 Hrs L + 15 Hrs P
L–T–P–J :	2–0–2-0		

COURSE VISION: This course develops the analytical skills, algorithmic thinking, learn C programming fundamentals to write clean, modular and optimized code. The learners can gain confidence in identifying errors and validating solutions for practical problems in mathematics, science and everyday scenarios using C. The course is extended to implement CUDA programming techniques to support parallel execution for massive datasets.

Course Learning Objectives:

This Course will enable students to:

1. Understand programming constructs of C language to solve the real problems
2. Explore simple data structures like arrays and strings to solve real world problems
3. Demonstrate modular codes using pointers and functional programming concepts
4. Demonstrate the concept that binds heterogenous data using structures and unions constructs
5. Introduce Nvidia CUDA parallel code execution and device memory management methods

BROAD COURSE DESCRIPTION:

This course introduces students to the fundamentals of programming and computational problem-solving using the C language. It focuses on building strong skills in logical thinking, algorithmic design, and structured programming techniques to address real-world challenges.

Learners will gain hands-on experience in writing, compiling, debugging, and executing C programs while exploring essential concepts such as Data types and control structures, Arrays and pointers, Functions, structures, and unions, File operations and dynamic memory allocation techniques.

Beyond C programming, the course provides a foundation in CUDA programming, a powerful framework for harnessing the computational capabilities of NVIDIA GPUs. It includes the topics namely, Installation and setup of the NVIDIA CUDA Toolkit, Parallel execution of large-scale data operations, CUDA compiler usage, Memory management across CPU and GPU, Profiling CUDA programs to observe performance gains.

Through a blend of theory and practical exercises, the course nurtures the ability to translate everyday problems into computational models and implement efficient solutions

UNIT – I	09 Hours

BASICS AND OVERVIEW OF C

Introduction to Problem Solving using Algorithms and Flowchart: Key features of Algorithms: Sequence, Decision, Repetition with examples. Background, structure of C program, keywords, Identifiers, Data Types, Variables, Constants, Input / Output statements, Operators (Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associativity, Expression Evaluation, Type conversions. Conditional Branching Statements-if and switch statements, iterative statements (loops)-while, for, do-while statements, Loop examples, Nested loops, break, continue, go to statement.

Engineering Applications: Industrial cooling systems, Automotive Engine Control Units

UNIT – II	09 Hours

ARRAYS AND STRINGS

1D-Array: Definition, declaration and initialization of one-dimensional array, reading and writing array elements, Operations on 1D-Array: traversal, searching (Linear and Binary search), sorting (Bubble sort and Selection Sort). 2D-Array: Declaration and Initialization of two-dimensional arrays. Operations on

2D-Array: Matrix Operations - addition, subtraction, multiplication and transpose operations. Strings/Character Arrays: Definition, declaration, initialization, and representation of strings, Input/Output operations on strings, String handling functions and character handling functions. **Engineering Applications:** Multi-Sensor Monitoring Dashboard, True Caller Simulation.	
UNIT – III	**09 Hours**
POINTERS AND USER DEFINED FUNCTIONS Pointers: Definition, declaration and initialization of pointers. Accessing values using pointers. Accessing array elements using pointers, Double pointers. User defined functions: Built-in and user defined functions, Definition, declaration and call of user-defined functions. Categories of functions with example. Pointers as function arguments, Array as function argument and return value, Pass-by-value and Pass-by-address, Recursion. **Engineering Applications:** Banking transaction systems, Sensor data Analysis.	
UNIT – IV	**09 Hours**
DYNAMIC MEMORY ALLOCATION USING CUDA Dynamic Memory allocation in C programs: Dynamic memory allocation, Host memory allocation process, allocating a block of memory, releasing the used space, altering the size of allocated memory. Introduction to CUDA: Heterogeneous Computing, Simple processing flow, CUDA toolkit installation, NVCC, Device memory management, Parallel program execution, Blocks, Threads and Indexing concepts. Engineering Applications: Operating system workflow, Real-time environmental monitoring systems.	
UNIT – V	**09 Hours**
STRUCTURES, UNIONS AND FILES Structures: Purpose and usage of structures. Declaration of structures. Assignment with structures. Structure variables and arrays. Nested structures. Student and employee database implementation using structures. Unions: Declaration and initialization of a union. Difference between structures and unions. Example programs. Files: Defining, open, read, write, seek and closing of both textual and random files. **Engineering Applications:** Student/Employee records, Memory optimization in embedded systems, Configuration management.	

AI Integration (AI Pedagogy)

AI Integration:

AIM: Understand how foundational Neural networks makes decisions using **input → weighted sum → activation → output** mirrors AI logic.

Example: Implement the foundational perceptron learning algorithm to represent the AND Logic Gate using Arrays, Loops and Functions:

a. Develop a Weighted sum function that multiplies inputs by weights and sums them.
b. Test with sample inputs and weights.
c. Develop a threshold activation and connect it to the weighted sum to produce binary outputs.
d. Write a loop that compares perceptron output to expected output. Adjust weights slightly if the output is wrong.
e. Run multiple iterations (epochs).
f.

Laboratory Component (1 Credit)

The following lab experiments can be carried out using **GCC, Nvidia CUDA Toolkit and Nvidia Nsight Visualization tool:**

Lab Experiments

1. Develop a program for an industrial cooling system that turns on a fan when temperature exceeds 70°C.
2. Develop a program for an automotive ECU that adjusts fuel injection based on RPM.
3. Create a program that simulates a True Caller application by matching input phone numbers with stored names.
4. Build a multi-sensor monitoring dashboard where commands are given as strings (SHOW, ALERT, RESET).
5. Write a program to access sensor data using pointers.
6. Implement double pointers to manage dynamic sensor arrays.
7. Create a function to calculate banking transaction balances.
8. Implement recursion to analyse sensor data trends.
9. Develop a code that allocate memory dynamically for storing environmental sensor data.
10. Develop a program to resize memory blocks as new data arrives.
11. Develop a simple Nvidia CUDA program to process large-scale environmental monitoring data.
12. Implement Nvidia CUDA parallel matrix multiplication to simulate operating system workflow.
13. Implement a student database using structures.
14. Create an employee record management system.
15. Implement reading and writing of sensor logs for configuration management.

AI-Integrated Lab Exercises (Indicative)

- **Exercise 1:** Make use of Nvidia CUDA Toolkit and Nsight systems to visualize the parallel execution cycle that adds 1 million elements.
- **Exercise 2:** Develop a CUDA program that performs element-wise addition of two arrays. The program should allocate memory on the GPU, transfer the input arrays from the host to the device, perform the addition in parallel using a CUDA kernel, and then transfer the result back to the host. Make sure to handle memory allocation and deallocation properly. Also visualize the same with Nvidia Nsight systems.

Assessment Scheme

Definitions

1. **"Internal Assessment"** refers to the Continuous Internal Assessment (CIA). The weightage of this component is **60%**. It includes:
   a. One **Mid Semester Examination (MSE)** (Descriptive-based),
   b. Six **Continuous Assessment Tool (CAT)**: It may include: Surprise test (Computer-based), Quiz, Project based learning, Reflection Note on Guest Lecture, Industrial visit, E-course certification, Building models, Group discussion, Case study, Seminar, or Open Book Assignment
   c. One **AI-powered Assignment (APA):** Use of AI tools for problem-solving while simultaneously developing Authentic Intelligence through critical evaluation, ethical reasoning, and responsible decision-making.

2. **“External Assessment”** refers to the Semester End Examinations (SEE). The weightage of this component is **40%**. At the end of semester, the final examination is conducted to evaluate student’s performance.

Integrated Professional Core Courses (Theory + Practical)

Criteria	**Weightage of Marks**
Continuous Assessment Tool (CAT) every 2 weeks (Minimum 05 assessments)	10
Mid Semester Examination (MSE)	20
Assignment on CUDA Programming	10
Lab Record	10
Lab Examination	10
Total	**60 marks**

Course Outcome	**Description**	**Bloom’s Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Apply fundamental programming constructs such as data types, operators, expressions, control flow, and loops to write basic C programs.	L3
CO2	Analyze the ability of one-dimensional and two-dimensional arrays to perform search, sort, matrix operations and manipulate strings using standard C functions.	L3
CO3	Develop functional codes with pointers including recursion, call-by-value/address, and arrays as arguments to build modular C programs.	L3
CO4	Develop programs using structures, unions and files to represent and manage compound data types for real-world applications like student or employee databases.	L3
CO5	Implement Nvidia CUDA dynamic memory allocation techniques for device memory management and parallel execution for massive datasets.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	1	2		1								1
CO2	2	2	2		1								1
CO3	2	2	2		1								1
CO4	2	2	2		2								1
CO5	2	2	2		1								1

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Recommended Tools

- Programming: GCC, NVCC
  - Visualization: Programiz Pro C Code Visualizer tool
  - AI Tools: Nvidia CUDA Toolkit, Nvidia Nvprof, Nsight systems

Suggested Reading

Primary Textbooks

1. Behrouz A. Forouzan, Richard F. Gilberg, “Computer Science - A Structured Approach Using C”, Cengage Learning, 2007.

Reference Textbooks

1. Reema Thareja, "Programming in C". Oxford University Press, Second Edition, 2016.
2. Brian W. Kernighan and Dennis M. Ritchie, (2012) “The C Programming Language”, 2nd Edition, PHI.
3. Vikas Gupta, “Computer Concepts and C Programming”, Dream tech Press 2013.

Additional Resources

1. NOC: Problem Solving Through Programming In C, IIT Kharagpur - NPTEL
2. C Tutorial
3. CUDA C/C++ BASICS - IIT Kanpur

DLI Course Mapped

An Even Easier Introduction to CUDA
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-AC-01+V1

DLI Learning Path

Accelerated Computing

****

Environmental Science

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code	:		**Credits**	: 02
Hours / Week	:	03 Hours	**Total Hours**	: 30 Hrs L
L–T–P-J	:	2–0–0-0		

Course Vision:
To equip engineering students with environmental science knowledge and AI-driven analytical skills to develop innovative, data-driven solutions for sustainable and resilient ecosystems.

Course Learning Objectives:
Students will:
1. Understand the evolution and diversity in nature.
2. Understand the concepts of environment, pollution, energy resources.
3. Learn water as a resource, rain water harvesting as a method of conversation of water
4. Understand solid waste and its management
5. Learn environmental Protection Act laws, environmental Impact Analysis and air Monitoring

BROAD COURSE DESCRIPTION
Environmental Science for Engineers is a multidisciplinary course that focuses on ecosystems, pollution, energy resources, disaster management, and environmental regulations along with their engineering applications. The course emphasizes environmental sustainability, resource management, and pollution control through practical and real-world approaches. By the end of the course, students will be able to understand environmental challenges and develop sustainable engineering solutions for society and industry.

Detailed Course Syllabus
The course consists of five modules. Each module integrates:
- Core topics
- Engineering applications
- AI integration
- Authentic Intelligence reflection

UNIT – I	**06 Hours**

ECOSYSTEMS and BIODIVERSITY
Topics: Ecosystem – structure and function – Producers, consumers, decomposers – Energy flow in ecosystem – Food chains, food webs, ecological pyramids – Environment: definition, scope and importance – Pollutants and xenobiotics (natural & anthropogenic) – Eco-kinetic and bio-kinetic properties – Dose-response relationships – 3 T's (Time, Type, Threshold) – Chronic and acute health effects – Sustainable Development Goals (SDGs).
Engineering Applications:
- Monitoring and managing ecosystems
- Biodiversity conservation and restoration
- Assessing pollutant impacts on humans and wildlife
- Environmental planning and sustainability studies
- Addressing SDG 15 – Life on Land

UNIT – II	**06 Hours**

POLLUTION AND MANAGEMENT
Topics: Air pollution and criteria pollutants – Ozone (O_3), Particulate Matter (PM), Carbon Monoxide (CO), Nitrogen Oxides (NOx), Sulphur Dioxide (SO_2), Lead (Pb) – Acid rain cycle – Greenhouse effect and global warming – Water resources: lentic and lotic systems – Rainwater harvesting – Water pollution and management strategies – Noise pollution: sources and effects – Municipal solid waste, hazardous waste, electronic waste, biomedical waste – Solid waste management techniques: landfills and composting.
Engineering Applications:

- Monitoring air, water, and soil quality
- Planning sustainable water and waste management systems
- Reducing noise and environmental hazards in urban and industrial areas
- Ensuring safe disposal of hazardous and biomedical waste
- Addressing SDG 14 – Life Below Water, SDG 6 – Clean Water and Sanitation, SDG 11 – Sustainable Cities and Communities and SDG 13 – Climate Action

UNIT – III	06 Hours

ENERGY

Topics: Types of energy – Conventional energy sources: fossil fuels, coal, solar, wind – non-conventional energy sources: biofuels, biomass, biogas – Energy production, conversion and utilization – Renewable and sustainable energy sources and their role in environmental sustainability.

Engineering Applications:

- Renewable energy systems (solar panels, wind turbines, bioenergy plants)
- Energy efficiency and sustainability planning
- Develop and optimize systems for biofuel production (fermentation,
- transesterification, anaerobic digestion)
- Integrate biofuels into hybrid energy systems alongside solar, wind, and smart grids.
- Addressing SDG 7 – Affordable and Clean Energy and SDG 12 – Responsible
- Consumption and Production

UNIT – IV	06 Hours

DISASTERS AND MANAGEMENT

Topics: Environmental and societal impacts of energy production – Ethical decision-making in energy planning and consumption – Balancing sustainability with technological and economic considerations.

Engineering Applications:

- Understanding environmental and societal impacts of energy production
- Ethical decision-making in energy planning and consumption
- Balancing sustainability with technological and economic considerations

UNIT – V	06 Hours

ENVIRONMENTAL IMPACT

Topics: Environmental Impact Assessment (EIA) – purpose, process and importance – Air pollution monitoring and Ambient Air Quality Standards (AAQS) – Environmental legislation: Environment Protection Act, 1986; Water (Prevention and Control of Pollution) Act; Wildlife Protection Act; Forest Conservation Act – Issues in enforcement of environmental laws – Public awareness and participation in environmental protection.

Engineering Applications:

- Planning and regulating developmental project
- Environmental monitoring and compliance
- Policy-making for sustainable development
- Educating the public on environmental responsibility

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Taxonomy Level
CO1	Critically elucidate the basic concepts that govern environmental quality, ambient air quality standards	L4

CO	Outcome	Bloom's Taxonomy Level
CO2	Compare different Energy resource and their environmental implications.	L4
CO3	Identify different types of pollution, waste streams.	L1
CO4	Identify different natural and manmade disasters and prevention.	L2
CO5	Apply the process of environmental impact assessment and implications of Indian Environment Laws	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	3	-	-	-	-	-	-	-	-	3	1	2
CO2	3	3	-	-	-	-	-	-	-	-	3	1	2
CO3	2	2	-	-	-	-	-	-	-	-	2	1	1
CO4	2	2	-	-	-	-	-	-	-	-	2	1	1
CO5	3	3	-	-	-	-	-	-	-	-	3	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

- **AI Integration:**
  - **Module-1:** This module integrates AI-based environmental monitoring systems to support analysis of ecosystem patterns and biodiversity datasets. It enables predictive modelling of environmental changes and trends, helping in data-driven interpretation of ecological systems.
  - **Module-2:** This module integrates AI-powered sensors for real-time monitoring of air, water, and soil quality, enabling continuous environmental data collection. It also uses remote sensing technologies for tracking large-scale environmental parameters and predictive models to analyse pollution trends and assess environmental risks. Additionally, AI-based decision support systems assist in efficient waste and resource management.
  - **Module-3:** This module integrates AI technologies in modern electricity systems for smart grid management, enabling predictive energy distribution based on demand patterns. It supports AI-assisted optimization of renewable energy output from sources such as solar and wind, and uses predictive maintenance techniques to monitor and improve the performance of power plants and energy infrastructure.
  - **Module-4:** This module integrates AI techniques for analysing satellite and climate data to support disaster prediction and environmental forecasting. It enables prediction of solar radiation, wind speed, and extreme weather events using data-driven models. It also incorporates AI-based early warning systems and predictive risk assessment tools for scenario simulation and hazard preparedness.
  - **Module-5:** This module integrates AI-based modelling techniques for predictive environmental

impact assessment of development projects. It supports real-time monitoring of air and water quality using intelligent systems and enables data-driven decision support for regulatory compliance. It also includes simulation tools to evaluate possible environmental outcomes of engineering and industrial activities.

- **Authentic Intelligence Reflection:**
  - **Module-1:** This module reflects authentic intelligence by enabling students to interpret ecosystem and pollution data in a responsible manner, ensuring accurate understanding of environmental conditions. It develops awareness of human impacts on biodiversity and ecological balance. It also promotes ethical and informed decision-making for environmental protection, encouraging sustainable practices and responsible engineering actions.
  - **Module-2:** This module reflects authentic intelligence by enabling students to interpret environmental data ethically and accurately for meaningful analysis. It develops understanding of the long-term impacts of pollutants on ecosystems and human health, supporting deeper awareness of environmental consequences. It also promotes informed decision-making in sustainable urban and industrial planning, encouraging responsible and future-oriented engineering solutions.
  - **Module-3:** This module reflects authentic intelligence by enabling students to understand the environmental and societal impacts of energy production systems. It develops ethical decision-making skills in energy planning and consumption, ensuring responsible use of resources. It also encourages balancing sustainability with technological advancement and economic feasibility, promoting holistic and informed engineering solutions for energy management.
  - **Module-4:** This module reflects authentic intelligence by enabling students to make ethical and responsible decisions in disaster management scenarios. It develops an understanding of the human and environmental consequences of natural and man-made disasters. It also encourages prioritizing safety, preparedness, and sustainability while designing effective response and mitigation strategies for disaster risk reduction.
  - **Module-5:** This module reflects authentic intelligence by enabling students to make ethical and informed decisions in environmental management practices. It develops the ability to balance developmental needs with ecological conservation, ensuring sustainable use of natural resources. It also promotes public awareness and responsibility toward environmental protection, encouraging active participation in building a sustainable and eco-friendly society.

Research Integration Teaching

Students review at least one topic:

- AI in climate change prediction
- Smart cities and sustainable development
- AI in biodiversity conservation
- AI-based pollution monitoring
- Renewable energy optimization using AI

Students present a **short research review**.

Assessment Scheme

Definitions

2. **“Continuous Internal Assessment** (CIA)**”** with a weightage of **60%**. It includes:
   a. **Mid Semester Examination (MSE)**: One descriptive examination to evaluate conceptual understanding of AI fundamentals and problem-solving approaches.
   b. Continuous Assessment Tools (CAT – Six Components): Includes

activities such as quizzes, surprise tests, case studies, group discussions, project-based learning, seminars, industrial visits, reflection notes, model building, e-course certifications, or open book assignments.

A. Professional Core Courses (Theory)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) – Quiz, Assignments, Write ups	60
Mid Semester Examination (MSE)	40
Total	**100 marks**

Recommended Tools

- Atmo-→ High-accuracy weather and climate prediction for disaster management
- Envirosuite-→ Real-time air, water, and noise pollution monitoring
- Google Earth Engine→ Satellite data analysis for deforestation, land use, climate studies
- iNaturalist→ AI-based species identification
- eBird→ Tracks bird populations and migration
- MegaDetector→ Detects animals using camera trap images
- Climate FieldView→ AI-based crop monitoring and yield prediction
- IBM Watson→ Used for environmental risk prediction and sustainability analytics
- AMP Robotics→ Automated waste sorting and recycling
- Kayrros Methane Watch→ Tracks methane emissions globally

Suggested Reading

1. Benny Joseph (2005). “Environmental Studies”, Tata McGraw – Hill Publishing Company Limited, New Delhi.
2. R. J. Ranjit Daniels and Jagadish Krishnaswamy (2014). “Environmental Studies” (2014), Wiley India Pvt Limited, New Delhi.
3. P. Aarne Vesilind, Susan M. Morgan, Thomson (2008). “Introduction to Environmental Engineering” (2008), Thomson learning, Second Edition, Boston.
4. R. Rajagopalan (2005). “Environmental Studies – From Crisis to Cure” Oxford University Press, New Delhi.
5. Zhou, K., Fu, C., & Yang, S. (2016). Big data driven smart energy management. Renewable and Sustainable Energy Reviews.

****

INDIAN CONSTITUTION AND LAW

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I / II

Subject Code :		**Credits**	: 01
Hours / Week :	03 Hours	**Total Hours**	: 15 Hrs L
L–T–P-J :	1–0–0-0		

Course Vision

The study of the Constitution of India helps engineering students understand the basic principles governing society, governance, and individual rights. It also nurtures ethically responsible engineers by creating legal awareness and encouraging professional integrity, social responsibility, and respect for constitutional values.

Prerequisites: Basic understanding of Indian society and governance, awareness of constitutional values, fundamental rights and duties, legal and ethical reasoning, and general knowledge of contemporary issues related to technology, digital rights, and Artificial Intelligence.

Course Learning Objectives:

This Course will enable students to:

1. Understand the basic principles of the Constitution of India.
2. Explain fundamental rights, duties, and directive principles.
3. Apply legal concepts to real-world situations.
4. Analyse the role of law in governance and society.
5. Develop awareness of ethical and legal responsibilities.

BROAD COURSE DESCRIPTION

This course provides a comprehensive introduction to the Constitution of India and the basic principles of law that govern society and the nation. It explains the fundamental features, structure, and functioning of the Indian legal and constitutional system, including Fundamental Rights, Duties, Directive Principles, and the role of the Judiciary. The course also explores the importance of constitutional values such as justice, equality, liberty, and secularism in maintaining democracy and social harmony. Further, it examines the relevance of legal awareness and ethical responsibility in professional and social life, especially for students and future professionals.

Detailed Course Syllabus

The course consists of five modules. Each module integrates:

- M Core topics
- Engineering applications
- AI integration
- Authentic Intelligence reflection

UNIT – I	06 Hours

FOUNDATIONS OF THE INDIAN CONSTITUTION

Topics: Meaning of the Constitution and Constitutionalism – Indian Constitution and its Salient Features – Sources of the Indian Constitution and Global Constitutional Influences – The Preamble of the Constitution.

Engineering Applications:

- Interpreting the Preamble and constitutional values in governance and public policy
- Relating constitutional principles to ethical engineering practices
- Evaluating laws and policies in technology-driven environments
- Understanding the role of AI integration within constitutional frameworks

UNIT – II	06 Hours

FUNDAMENTAL RIGHTS, DIGITAL RIGHTS & DUTIES

Topics: Importance and Features of Fundamental Rights – Types of Fundamental Rights – Right to Privacy, Data Protection and Digital Rights in the AI Era – Directive Principles of State Policy – Fundamental Duties.

Engineering Applications:

- • Protecting individual rights in real-life and digital environments
  • Understanding data privacy, cybersecurity, and digital rights in the AI era
  • Applying Directive Principles in governance and public welfare systems
  • Promoting justice, equality, and ethical responsibility in technological innovation

UNIT – III	06 Hours

JUDICIARY, TECHNOLOGY AND JUDICIAL REVIEW

Topics: Independence of the Judiciary in Democratic Governance – Judicial Review and Judicial Activism – Role of Courts in Regulating AI and Digital Platforms – Public Interest Litigation.

Engineering Applications:

- • Applying judicial review in regulating AI and emerging technologies
  • Understanding legal accountability of digital platforms and online systems
  • Using Public Interest Litigation to address social and technological issues
  • Promoting justice and rights protection in a digital society

UNIT – IV	06 Hours

AMENDMENT AND EMERGENCY PROVISIONS

Topics: Amendment of the Constitution: Power and Procedure – Historical Perspective of Constitutional Amendments in India – Emergency Provisions – National Emergency – State Emergency – Financial Emergency.

Engineering Applications:

• Understanding constitutional changes affecting technology and governance
• Evaluating emergency measures in digital and cyber-security contexts
• Analysing the role of constitutional safeguards during national crises
• Studying the balance between state power and individual rights

UNIT – V	06 Hours

INTELLECTUAL PROPERTY RIGHTS, INNOVATION & AI

Overview and Importance of Intellectual Property Rights – Importance of Human Creativity in the Present Scenario – Patents – Trademarks – Copyrights – Artificial Intelligence and Emerging Technologies in Innovation.

Engineering Applications:

- • Protecting technological innovations through patents, trademarks, and copyrights
  •Encouraging ethical and creative use of AI technologies
  •Applying IPR laws in engineering and research innovations
  • Managing ownership, originality, and legal rights in AI-driven creations

COURSE OUTCOMES (CO)

CO	Outcome	Bloom's Levels
CO1	Outline the structure and features of the Constitution of India	L2
CO2	Interpret key constitutional and legal concepts	L2

CO	Outcome	Bloom's Levels
CO3	Examine the functioning of government institutions	L3
CO4	Relate legal principles to engineering and societal issues	L3
CO5	Assess ethical, legal, and professional responsibilities	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	-	-	-	-	-	-	-	-	2	1	1
CO2	2	2	-	-	-	-	-	-	-	-	2	1	1
CO3	3	3	-	-	-	-	-	-	-	-	2	1	2
CO4	3	3	-	-	-	-	-	-	-	-	3	1	2
CO5	3	3	-	-	-	-	-	-	-	-	3	1	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

AI INTEGRATION & AUTHENTIC INTELLIGENCE REFLECTION (AI² PEDAGOGY)

- **AI Integration:**
- **Module-1:** This module reflects AI integration by helping students understand the foundations of the Indian Constitution, constitutional values, and governance principles, while analysing the role of AI-driven policies, digital governance, and constitutional interpretation in modern society.
- **Module-2:** This module reflects AI integration by emphasizing the protection of fundamental rights, digital rights, privacy, and data protection in the AI era, enabling students to understand ethical and legal challenges arising from artificial intelligence and emerging technologies.
- **Module-3:** This module reflects AI integration by focusing on the role of the judiciary and judicial review in regulating AI systems, digital platforms, and online governance, while promoting accountability, transparency, and protection of rights in technology-driven environments.
- **Module-4:** This module reflects AI integration by encouraging students to examine constitutional amendments, emergency provisions, and governance mechanisms in relation to cyber security, digital emergencies, and technology-based administrative systems.
- **Module-5:** This module reflects AI integration by promoting awareness of Intellectual Property Rights, innovation, and AI-based creations, enabling students to understand legal ownership, ethical use of technology, patents, copyrights, and trademarks in the digital age.

- **Authentic Intelligence Reflection:**
- **Module-1:** This module reflects authentic intelligence by encouraging students to understand constitutional values such as justice, liberty, equality, and fraternity, and apply them responsibly in social, professional, and technological contexts.

- **Module-2:** This module reflects authentic intelligence by promoting ethical thinking, protection of human dignity, responsible use of digital technologies, and awareness of individual duties and social responsibilities in democratic society.
- **Module-3:** This module reflects authentic intelligence by developing critical thinking regarding judicial independence, judicial activism, and public interest litigation, helping students evaluate social and technological issues through legal reasoning.
- **Module-4:** This module reflects authentic intelligence by helping students analyze constitutional amendments and emergency powers critically, ensuring balanced understanding between national security, governance, and protection of citizens' rights.
- **Module-5:** This module reflects authentic intelligence by encouraging creativity, innovation, and ethical responsibility in the use of Artificial Intelligence and emerging technologies, while respecting intellectual property and human contribution.

Research Integration Teaching

Students review on ***at least one recent research topics*** such as:

Data Privacy and Digital Rights in the AI Era
• Role of Judiciary in Regulating Artificial Intelligence
• Cyber Laws and Constitutional Protection in Digital Society
• Intellectual Property Challenges in AI-Generated Content
• Ethical and Legal Issues in Artificial Intelligence and Emerging Technologies
• Judicial Activism and Public Interest Litigation in Technology Governance
• AI-Based Surveillance and Right to Privacy

Students present a **short research review**.

Assessment Scheme

Definitions

1. **"Internal Assessment (IA)"** refers to the Continuous Internal Assessment (CIA) with a weightage of 60%. It includes:
   a. **Mid Semester Examination (MSE):** One descriptive examination to evaluate students' understanding of constitutional principles, legal concepts, governance, and contemporary legal issues related to technology and society.
   b. **Continuous Assessment Tools (CAT – Six Components):** Includes activities such as quizzes, surprise tests, case studies, group discussions, seminars, project-based learning, reflection notes, presentations, assignments, constitutional case analysis, or open book assignments.

A. Professional Core Courses (Theory)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) – CBT	30 Marks
Assignment	05 Marks
Notes Submission	05 Marks
Presentation / Seminar	10 Marks
Total	**50 Marks**

Suggested Reading

Primary Textbooks

1. Introduction to the Constitution of India by D. D. Basu, 25th edition, 2021 published by LexisNexis
2. Indian Constitutional law by M. P. JAIN, 8th edition, 2022 published by LexisNexis
3. The Indian Constitution: Cornerstone of a nation by Granville Austin, 2nd edition, 2000 published by Oxford University Press.

Additional Resources

1. Gautam Bhatia: The Indian Constitution: A Conversation with Power. Oxford University Press, 2019.

2. Purvi Pokhariyal, Amit K. Kashyap, and Arun B. Prasad: Artificial Intelligence: Law and Policy Implications. Eastern Book Company, 2021.
3. P. Narayanan: Intellectual Property Law, Latest Edition 2017 Eastern Law House, Latest Edition.

****

Applied Statistics & Stochastic Processes

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - III

Course Code	**:**		**Credits**	**:**	03
Hours / Week	**:**	03	**Total Hours**	**:**	30 Hrs L
L–T–P–J	**:**	3–0–0–0			

Course Vision:
To develop strong foundations in probability, statistical inference, and stochastic process, enabling students to analyze uncertainty, make data-driven decisions, and explore advanced computational models including AI systems.

Course Objectives:
This course will enable students to:
1. Understand the concepts of random variables and probability distributions for modelling uncertainty.
2. Analyze relationships between variables using correlation and regression techniques.
3. Apply statistical inference methods for decision-making under uncertainty.
4. Use parametric and non-parametric tests for hypothesis testing and data analysis.
5. Understand the fundamentals of stochastic processes and their Applications in modeling random systems, decision-making, and AI-driven Applications.

Broad Course Description: This course provides a comprehensive foundation in probability theory, statistical inference, and stochastic processes for analyzing uncertainty and supporting data-driven decision-making. It covers key concepts such as random variables, probability distributions, correlation, regression, and both parametric and non-parametric hypothesis testing. The course further introduces statistical inference techniques for drawing meaningful conclusions from data and explores stochastic processes for modeling dynamic random systems. Through theoretical concepts and practical Applications, students gain the ability to apply statistical and probabilistic methods in real-world domains such as engineering, finance, and AI-driven systems.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various

Course Outcomes:
1. Lecture Method**:** Concepts, algorithms, and models are explained through structured lectures supported by R Programming demonstrations.
2. Interactive Teaching**:** Active learning is promoted through discussions, brainstorming, and group problem-solving activities.
3. Demonstration using R Programming and Python**:** AI techniques like classification and optimization are demonstrated to visualize results and understand implementation.
4. Multimedia Learning: Videos and simulations are used to explain complex concepts and real-world Applications.
5. Collaborative Learning: Students work in groups to implement projects and analyze datasets using computational tools.
6. Higher Order Thinking Questions: Analytical questions are used to develop model design, evaluation, and decision-making skills.
7. Problem-Based Learning: Real-world problems are given to apply techniques using Python

for practical solutions.

Unit – I	**09 Hours**
Random Variables and Probability Distributions	
Topics: Introduction to random variables, probability mass function (PMF), probability density function (PDF), distribution functions, joint probability distribution and joint density functions,	

marginal and conditional distributions, mathematical expectation and its properties, covariance, moment generating function, characteristic function. **Applications:** Risk analysis and financial modelling, Reliability engineering, Data analysis and forecasting, Signal processing	
Unit – II	**09 Hours**
Bi-variate Data, Correlation and Regression Analysis **Topics:** Bi-variate random variables, correlation, types of correlation, measures of correlation, graphical methods (scatter diagram), Karl Pearson's correlation, rank correlation (for repeated and non-repeated ranks), linear regression model, regression lines, least squares method, fitted model. **Applications:** Market trend analysis, Economic forecasting, Quality control and process analysis, Data relationship analysis	
Unit – III	**09 Hours**
Inference for Decision Making – I **Topics:** Introduction to hypothesis, hypothesis testing procedure, p-values and decision making, test of hypothesis for proportion (single and difference of proportions), testing hypothesis about mean (Z-test and t-test), two-sample tests, test for equality of variance (F-test), ANOVA (one-way analysis of variance). **Applications:** Quality assurance and control, Clinical trials and testing, Business decision-making, Industrial process evaluation	
Unit – IV	**09 Hours**
Inference for Decision Making – II **Topics** Introduction to non-parametric tests, Chi-Square test for goodness of fit, Chi-Square test using probability distributions, Chi-Square test for independence of attributes. **Applications:** Survey data analysis, Market research, Pattern recognition, Categorical data analysis	
Unit – V	**09 Hours**
Stochastic Processes **Topics:** Introduction to Stochastic Processes (SPs), Markov chains, Poisson process, Discrete-time Markov Chains (DTMCs), Continuous-time Markov Chains (CTMCs), Monte Carlo Simulation. **Applications** Queueing systems and waiting line models, Reliability and maintenance systems, Population growth and epidemic modelling, Financial market analysis	

Authentic Intelligence:

1. Understanding uncertainty in AI predictions
2. Avoiding misleading interpretations of correlations
3. Interpreting statistical results responsibly
4. Ethical handling of categorical data
5. Ethical use of simulation-based decision models

AI Integration:

- Feature distribution modelling in data science
- Regression models in machine learning
- Statistical decision-making in AI systems
- Model validation using statistical tests
- Markov models in reinforcement learning and decision processes

Course Outcome

Course Outcomes	Description	Bloom's Taxonomy Level

(COs)		
At the end of the course the student will be able to:		
1	Apply probability concepts and random variables to analyze uncertainty	L3
2	Analyze data using correlation and regression techniques	L4
3	Perform hypothesis testing and statistical inference for decision making	L4
4	Apply non-parametric tests for categorical data analysis	L3
5	Model and analyze stochastic processes such as Markov chains	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

	Program Outcomes (POs)											**PSOs**	
COS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO-1	3	2			2					1		2	–
CO-2	3	3	2	2	3				1	1		3	2
CO-3	3	3	3	2	3				1	2		3	2
CO-4	3	2	2	2	2					1		2	2
CO-5	3	3	2	3	3			1		1	2	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Mendenhall, William, Robert J. Beaver, and Barbara M. Beaver. *Introduction to probability and statistics*. Cengage Learning, 2020.
2. Ross, Sheldon, M. (1995). *Stochastic Processes* (2nd edition), John Wiley.

REFERENCE BOOKS:

1. R.E. Walpole, R.H. Mayers, S.L. Mayers and K.Ye, Probability and Statistics for engineers and scientists, 9th Edition, Pearson Education, 2018.
2. Douglas A. Wolfe, Grant Schneider, Intuitive Introductory Statistics, Springer, 2017.
3. Miller & Freund's, Probability and statistics for engineers, 8th edition, Pearson publication, 2018.
4. Richard I. Levin, David S. Rubin, Masood H. Siddiqui, Sanjay Rastogi, Statistics for Management, 8th Edition, Pearson Publications, 2018.
5. S. C. Gupta and V. K. Kapoor, Fundamentals of Applied Statistics, S. Chand, 2006.
6. Mark L. Berenson, David M. Levine, Kathryn A. Szabat, Basic Business Statistics Concept and Application, Thirteenth Edition, Pearson Education Limited, 2015.
7. Bruce L. Bowerman, Richard T. O'Connell, Emily S. Murphree, Business Statistics in Practice, Seventh Edition, McGraw-Hill, 2014.
8. Douglas C. Montgomery, George C. Runger, Applied Statistics and Probability for Engineers, Sixth Edition, John Wiley and Sons, 2014.
9. Norean R. Sharpe, Richard D. De Veaux, Paul F. Velleman, Business Statistics, Fourth Edition, Pearson Education. 2019Trivedi, K. S. (2016), Probability &

Statistics with Reliability, Queuing & Computer Science Applications (2nd edition), New Jersey, John Wiley & Sons, Inc.

RECOMMENDED TOOLS

- Python
- R Programming
- NumPy
- Pandas
- Matplotlib
- Scikit-learn

LAB EXERCISES

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Random Variables and Probability Distributions (03 Hours)					
1	Random Variable Visualization	Generate and visualize discrete and continuous random variables	R Programming	1 h	**CO1**
2	PMF, PDF & CDF	Compute PMF, PDF, and CDF for standard distributions	R Programming	1 h	**CO1**
3	Joint Distributions	Simulate joint probability distributions	R Programming	1 h	**CO1**
MODULE II: Exercises 4–6: Bi-variate Data, Correlation and Regression Analysis (03 Hours)					
4	Bivariate Data Visualization	Visualize bivariate data using scatter plots	R Programming	1 h	**CO2**
5	Correlation Analysis	Compute Karl Pearson's and Spearman's rank correlation	R Programming	1 h	**CO2**
6	Linear Regression	Fit and evaluate regression model	R Programming	1 h	**CO2**
MODULE III: Exercises 7–9: Inference for Decision Making – I (03 Hours)					
7	Z-Test & t-Test	Perform Z-test and t-test (single & two-sample)	R Programming	1 h	**CO3**
8	Proportion Testing	Test for proportion and difference of proportions	R Programming	1 h	**CO3**
9	F-Test & ANOVA	Perform F-test and one-way ANOVA	R Programming	1 h	**CO3**
MODULE IV: Exercises 10–12: Inference for Decision Making – II (03 Hours)					
10	Chi-Square Tests	Perform goodness of fit and independence tests	R Programming	1 h	**CO4**
11	Bayesian Inference	Implement basic Bayesian inference techniques	R Programming	1 h	**CO4**
12	Data Visualization	Visualize probability distributions and statistical results	R Programming	1 h	**CO4**
MODULE V: Exercises 13–15: Stochastic Processes (03 Hours)					
13	Monte Carlo Simulation	Perform Monte Carlo simulations for probabilistic models	R Programming	1 h	**CO5**
14	Markov Chains	Construct transition matrices and simulate state transitions	R Programming	1 h	**CO5**
15	Exponential Distribution	Study exponential distribution and waiting time analysis	R Programming	1 h	**CO5**

****

Logic Design & Computer Organization Architecture

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:		**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45 Hrs L
L–T–P–J	:	3–0–0–0			

Course Vision:

The course aims to develop strong foundational knowledge in digital logic design and computer organization to help students understand, analyze, and design efficient computing systems and modern processor architectures. It equips students with the principles of Boolean algebra, combinational and sequential circuit design, memory organization, and CPU architecture while enhancing analytical and problem-solving skills required for embedded systems, intelligent computing, and advanced computer engineering applications.

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamental concepts of Boolean algebra, logic gates, and combinational circuit design.
2. Simplify and implement digital circuits using standard techniques (K-map, Quine–McCluskey).
3. Design and analyze sequential circuits using flip-flops, counters, and registers.
4. Explain the basic structure and operation of computer organization components such as ALU, memory, and control unit.

Apply logic design principles to understand processor organization and performance optimization in modern architectures.

Broad Course Description

The Logic Design and Computer Organization course provides a comprehensive understanding of how digital systems and computers process information. It begins with an introduction to number systems, logic gates, and Boolean algebra, progresses into combinational and sequential logic circuit design, and culminates in the study of computer architecture components—including CPU organization, memory hierarchy, and I/O systems.

The course bridges digital logic theory with how real-world processors and embedded systems operate. Through simulation, circuit design, and AI-assisted hardware emulation tools, students gain the skills necessary to design and evaluate efficient computing architectures.

UNIT – I	09 Hours

NUMBER SYSTEMS AND BOOLEAN ALGEBRA:

Number systems: Binary, Octal, Hexadecimal conversions; Binary arithmetic (Addition, Subtraction, Multiplication, Division)

Boolean algebra: Axioms, theorems, DeMorgan's laws

Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR

Standard forms: SOP and POS expressions, Minimization using Boolean laws and K-map (up to 5 variables), Logic circuit implementation using basic and universal gates

(Text Book 1: Chapters 1, 2)

UNIT – II	09 Hours

COMBINATIONAL LOGIC CIRCUITS:

Design and simplification using K-maps, Adders, subtractors, encoder, decoder, multiplexer, demultiplexer, Comparators, parity generator/checker, Design examples of combinational blocks for arithmetic processing, Implementation using MSI and LSI circuits

(Text Book 1: Chapters 3, 4)

UNIT – III	09 Hours

SEQUENTIAL LOGIC CIRCUITS:

Latches and Flip-Flops: SR, JK, D, T types, Conversion between flip-flop types, Excitation tables and state equations, Counters and registers: asynchronous/synchronous, Johnson, Ring counters, Sequential circuit design methodology (state diagrams, state tables), Applications of flip-flops in timing and control circuits (Text Book 1: Chapter 5; Text Book 2: Chapter 8)	
UNIT – IV	**09 Hours**
COMPUTER ORGANIZATION BASICS: Basic structure of computer: CPU, memory, I/O structures, Representation of instructions and instruction cycle, Hardwired vs microprogrammed control, Arithmetic and Logic Unit (ALU) design, Data path design and control signals, Stack organization and address sequencing, Interrupts and handling mechanisms. (Text Book 2: Chapters 10, 11, 12)	
UNIT – V	**09 Hours**
MEMORY, I/O, AND PERFORMANCE: Memory hierarchy: primary, secondary, cache, Cache mapping techniques (direct, associative, set-associative) Memory interleaving and virtual memory concepts, I/O methods: Programmed, Interrupt-driven, DMA, Bus organization and data transfer mechanisms, Introduction to pipelining and parallelism in CPUs (link to modern GPUs) (Text Book 2: Chapters 13, 14, 15)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **Sustainable Hardware & Green Computing:** Optimize logic circuits for reduced power usage and efficient hardware design.
- **Edge-AI and Embedded Systems:** Apply combinational and sequential circuit knowledge for efficient AI deployment at the edge.
- **Hardware-Aware AI:** Connect logic and memory design principles with GPU architectures (NVIDIA Ampere, Hopper).
- **Digital Twins & Simulation:** Use HDL simulators and AI-based predictive modeling for system performance estimation.
- **Learning Acceleration:** Explore parallelism and pipelining principles foundational to AI accelerators (CUDA-based GPUs).

Applications

1. Smart hardware controllers and embedded systems
2. Digital signal processing units in robotics
3. ALU design for neural computation units
4. Pipeline design for AI-based CPUs/GPUs
5. Memory organization for high-speed inferencing

Mini Project Component

Project 1 — CPU Design Simulator

- *Problem:* Design a simple CPU with instruction set and ALU operations.
- *Concepts used:* Sequential circuits, ALU design, control logic.
- *Industry link:* Hardware simulation (NVIDIA Isaac / Jetson).

Project 2 — Cache Mapping Performance Analyzer

- *Problem:* Create a tool to compare different cache mapping techniques.
- *Concepts used:* Memory hierarchy, address mapping, timing simulation.
- *Industry link:* High-performance computing (NVIDIA CUDA optimization).

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Analyze number systems, Boolean algebra, and logic gates for digital circuit foundations.	L2
CO2	Design and simplify combinational logic circuits using standard techniques.	L3
CO3	Develop and analyze sequential circuits using flip-flops, counters, and registers.	L3
CO4	Explain the architecture and organization of computer system components.	L2
CO5	Apply logic and architecture principles for efficient hardware and AI computing designs.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	-									2	
CO2	2	3	3		2							3	
CO3	2	2	3	2	2							2	
CO4	3	2										2	
CO5	2	2	3		3							3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. M. Morris Mano and Michael D. Ciletti, Digital Design, 6th Edition, Pearson, 2019.
2. Carl Hamacher, Zvonko Vranesic, and Safwat Zaky, Computer Organization, 6th Edition, McGraw Hill, 2012

REFERENCE BOOKS:

1. William Stallings, Computer Organization and Architecture, 10th Edition, Pearson,

2016.David A. Patterson and John L. Hennessy, Computer Organization and Design: RISC-V Edition, Morgan Kaufmann, 2021.
2. Charles H. Roth & Larry Kinney, Fundamentals of Logic Design, 7th Edition, Cengage, 2014.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Fundamentals of Accelerated Computing with CUDA C/C++	Unit IV & V	Relates to processor, ALU, and parallel execution design.
Fundamentals of Deep Learning for Multi-GPU and CPU Systems	Unit V	Maps to pipelining, memory hierarchy, and performance optimization.
Edge AI and Embedded Systems with Jetson	Unit II & III	Links HDL design and sequential circuits for embedded systems.
Fundamentals of Accelerated Data Science	Unit IV	Links architecture and computation with high-performance GPU data handling.

E-RESOURCES:

1. nptel.ac.in
2. nptel.ac.in
3. nvidia.com
4. coursera.org
5. cse02-iiith.vlabs.ac.in

Activity-Based Learning (Suggested Activities in Class)

1. Hands-on Boolean simplification puzzles (group-based)
2. HDL simulation using online tools (e.g., Logisim, Quartus)
3. Circuit role play (e.g., flip-flop behavior via human sequence demonstration)
4. FPGA mini-lab: real-time digital signal demonstration
5. Flip classrooms on pipeline design using GPU analogy

****

Database Management System

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	04
Hours / Week	:	04 Hours	**Total Hours**	:	60 Hrs (45Hrs L +15Hrs P)
L–T–P-J	:	3–0–2-0			

Course Vision: This course aims to build a strong foundation in database systems by enabling students to understand how data is structured, stored, managed, and retrieved efficiently in modern computing environments. It aims to develop practical and conceptual skills in relational and non-relational database technologies, including SQL, NoSQL, and distributed databases. The course further aims to prepare learners to design robust, scalable, and secure data management solutions that support real-world applications in software systems, data engineering, and analytics.

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamental concepts of database systems, data models, and database architecture.
2. Learn relational algebra, SQL, and query processing techniques for data retrieval.
3. Study normalization techniques and transaction management including ACID properties.
4. Explore advanced concepts including NoSQL, distributed databases, and data warehousing.

Broad Course Description

The course Database Management Systems provides a comprehensive understanding of how data is organized, stored, and efficiently retrieved using structured query languages and relational models. It covers foundational concepts including ER modeling, relational algebra, SQL, normalization (1NF–BCNF), transaction management with ACID properties, concurrency control mechanisms (2PL, timestamp), and recovery techniques. Advanced topics include indexing (B+ Trees, Hashing), query optimization, NoSQL databases (MongoDB, CAP theorem), distributed databases, and data warehousing (OLAP, Star and Snowflake schemas, Hadoop basics). Students gain hands-on experience with industry-standard tools MySQL and MongoDB, building a solid foundation for data engineering and database administration roles.

UNIT – I	09 Hours

Introduction to DBMS: Database concepts and architecture, Data models – Hierarchical, Network, Relational. Entity-Relationship (ER) modeling – entities, attributes, relationships, cardinality, Enhanced ER. Mapping ER model to relational schema. (Text Book-1: Chapters 1–3)

Textbook Reference:

• Elmasri & Navathe, Fundamentals of Database Systems, 7th Ed – Ch. 1–3 (Pearson, 2016)

• Silberschatz et al., Database System Concepts, 7th Ed – Ch. 1–2 (McGraw-Hill, 2019)

UNIT – II	09 Hours

Relational Model & SQL: Relational model concepts, Integrity constraints, Relational Algebra – Select, Project, Join, Set operations. Relational Calculus. SQL – DDL, DML, DCL, TCL, Nested queries, Views, Stored Procedures, Triggers. (Text Book-1: Chapters 6–8)

Textbook Reference:

• Elmasri & Navathe – Ch. 6–8

• Ramakrishnan & Gehrke, Database Management Systems, 3rd Ed – Ch. 4–5

UNIT – III	09 Hours

Database Design & Normalization: Functional dependencies, Armstrong's Axioms, Attribute closure, Normal Forms – 1NF, 2NF, 3NF, BCNF, 4NF, 5NF. Decomposition – lossless join and dependency preservation. Multivalued dependencies. (Text Book-1: Chapters 14–15)

Textbook Reference:

• Elmasri & Navathe – Ch. 14–15

• Silberschatz et al. – Ch. 7

UNIT – IV	09 Hours
Transaction Management & Concurrency Control: ACID properties, Transaction states, Serializability, Lock-based concurrency control, Timestamp ordering, Optimistic concurrency, Two-Phase Locking (2PL), Deadlock detection and prevention, Recovery – log-based recovery, checkpointing, shadow paging. (Text Book-1: Chapters 20–21) Textbook Reference: • Silberschatz et al. – Ch. 15–17 • Elmasri & Navathe – Ch. 20–21	
UNIT – V	**09 Hours**
Advanced Database Concepts: Indexing – B+ Trees and Hashing. Query optimization. Distributed databases. NoSQL – Document, Key-Value, Column, Graph stores, MongoDB, CAP theorem. Data Warehousing – OLAP vs. OLTP, Star/Snowflake schemas. Introduction to Hadoop ecosystem. (Text Book-1: Chapters 11–12, 19, 22–23) Textbook Reference: • Silberschatz et al. – Ch. 11–12, 19 • Date, An Introduction to Database Systems, 8th Ed – Ch. 21	

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Unit 5	**Accelerate Data Science Workflows With Zero Code Changes**	Data Science / RAPIDS	2 Hrs	—	courses.nvidia.com/ course-v1:DLI+T-DS-03+V1	GPU-accelerated pandas/scikit-learn replacement via RAPIDS — directly supports NoSQL, OLAP, and large-scale data analytics topics in Unit 5. Students learn to accelerate DB-style queries without code changes.
Units 4–5	**Accelerating End-to-End Data Science Workflows**	Data Science / RAPIDS	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-DS-02+V2	End-to-end GPU data pipeline (cuDF, cuML) — maps to accelerated query processing, data warehousing workflows, and big data analytics in Units 4–5.
Unit	**RAPIDS Accelerator for Apache Spark**	Data Science			learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-DS-04+V1	RAPIDS Accelerator for Apache Spark — reinforces Hadoop ecosystem and

5			2 Hrs	—		distributed data processing topics in Unit 5 with GPU-accelerated ETL and ML.
Units 1–2	**Introduction to Graph**	Deep Learning	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id=	Graph Neural Networks for graph-structured relational
	Neural Networks				course-v1:DLI+S-FX-05+V1	data — extends ER modeling (Unit 1) and relational algebra (Unit 2) to modern graph DB representations relevant to NoSQL graph stores.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Query Optimization:** Use AI tools to analyze and suggest optimized SQL query execution plans; reflect on performance vs. hand-tuned queries.
- **Reflective Schema Design:** Critically compare AI-generated ER diagrams against manually designed schemas for completeness and correctness.
- **Natural Language to SQL:** Explore NL2SQL tools (Text-to-SQL) to understand query semantics and validate AI-generated SQL for correctness.
- **Intelligent Normalization:** Use AI tools to identify functional dependencies and evaluate normalization suggestions against theoretical rules.
- **Human–AI Collaboration in DBA:** Promote use of AI monitoring tools for DB performance while maintaining mandatory human oversight for critical systems.

Applications:

- Banking and Financial Systems
- E-Commerce and Inventory Management
- Healthcare Record Management (EHR)
- Supply Chain and Logistics Systems
- Social Media Analytics and Reporting

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Create databases using DDL commands; apply PK, FK, Unique, Check, Not Null constraints.
- Write DML queries with GROUP BY, HAVING, ORDER BY, and aggregate functions.
- Implement nested queries, correlated sub-queries, and set operations (UNION, INTERSECT, MINUS).
- Create Views, Sequences, and Synonyms in a relational database.

• Design and implement Stored Procedures and Functions with input/output parameters.

• Create BEFORE and AFTER Triggers for INSERT, UPDATE, DELETE events.

• Implement transaction control using COMMIT, ROLLBACK, and SAVEPOINT with concurrency scenarios.
• Design normalized schema (up to BCNF) from a case study with full justification.
• Implement B+ Tree and Hash indexing; compare query performance with and without indexes.
• Explore MongoDB: CRUD operations, aggregation pipeline, indexing, and schema design.

Mini Project Component

• **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – Library Management System

Problem: Complete library system with book catalog, member management, and issue/return tracking.

Concepts used: ER modeling, SQL, Triggers, Stored Procedures

Industry relevance: Academic institutions and public libraries.

Project 2 – Online Examination Portal

Problem: Database backend for exams: question banks, enrollment, result generation.

Concepts used: Normalization, Complex SQL queries, Views

Industry relevance: EdTech and competitive exam portals.

Project 3 – Hospital Management System

Problem: Patient records, doctor appointments, prescriptions, and billing database.

Concepts used: ER modeling, Transactions, ACID, Concurrency control

Industry relevance: Healthcare IT and EHR platforms.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand fundamental database concepts, data models, and entity-relationship modeling for database design.	L2
2	Apply relational model concepts and write complex SQL queries for data retrieval, manipulation, and administration.	L3
3	Design normalized relational schemas by identifying functional dependencies and applying normalization up to BCNF.	L3
4	Analyze and implement transaction management techniques including concurrency control and recovery mechanisms.	L4
5	Evaluate and apply advanced database technologies including NoSQL, distributed databases, and data warehousing.	L5

Table: Mapping Levels of COs to POs / PSOs													
	Program Outcomes (POs)											**Program Specific Outcomes (PSOs)**	
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO 1**	**PSO 2**
CO1	3	2	2	3	2						2	3	2
CO2	3	3	2	3	3						2	3	2
CO3	3	3	3	3	2						2	3	2
CO4	3	3	2	3	3						2	3	3
CO5	2	3	2	3	3						2	3	2
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools:

- MySQL / PostgreSQL
- MongoDB
- MySQL Workbench / pgAdmin
- XAMPP / WAMP
- Draw.io / Lucidchart

TEXT BOOKS:

1. Ramez Elmasri & Shamkant Navathe, Fundamentals of Database Systems, 7th Ed, Pearson, 2016.
2. Abraham Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts, 7th Ed, McGraw-Hill, 2019.
3. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, 3rd Ed, McGraw-Hill, 2003.

REFERENCE BOOKS:

1. C.J. Date, An Introduction to Database Systems, 8th Ed, Pearson, 2004.
2. Thomas Connolly & Carolyn Begg, Database Systems: A Practical Approach, 6th Ed, Pearson, 2015.

E-RESOURCES:

- https://nptel.ac.in/courses/106105175
- https://www.coursera.org/learn/database-management
- https://www.udemy.com/topic/sql/free/
- https://cse01-iiith.vlabs.ac.in/

Activity Based Learning (Suggested Activities in Class)

- Real-world problem solving via group discussion: Design a DB schema for a ride-sharing app.

- Role play – simulate a DB administrator approving concurrent transactions.
- Flip class: Arrays and pointers review; students teach SQL joins to the class.
- Demonstration: Students benchmark query performance before and after indexing.

****

Data Structures Using C

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs (30Hrs L+15 Hrs P)
L–T–P-J	:	2–0–2-0			

Course Vision:

The course aims to develop strong foundations in data structures and algorithmic problem-solving using the C programming language. It enables students to understand efficient data organization, storage, and retrieval techniques for computational applications. The course enhances logical thinking and programming skills required for designing optimized software solutions. It also prepares students for advanced studies in algorithms, software development, and emerging computing technologies.

Prerequisites: Basic Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. Learn the fundamental data structures and identify data structuring strategies that are appropriate for a given contextual problem.
2. Illustrate and implement basic data structures such as stack, queue and linked list and apply them for the given problem.
3. Understand and distinguish the conceptual and applicative differences in trees, binary trees and binary search trees. Apply the concepts of trees for the given application.
4. Create and use appropriate data structures in C programs for solving real life problems.

Broad Course Description

The course Data Structures using C provides a comprehensive understanding of how data is organized, stored, and manipulated efficiently using the C programming language. It covers fundamental concepts such as arrays for sequential storage, stacks and queues for linear data processing, and linked lists for dynamic memory management. The course further explores hierarchical structures like trees and heaps, along with advanced techniques such as hashing for fast data retrieval and graphs for modelling complex relationships. Overall, it equips students with the skills to design efficient algorithms and solve real-world computational problems.

UNIT – I	06 Hours

INTRODUCTION:

Review of Pointers and Dynamic Memory Allocation, Structures: Types of Structures, Array of Structures, Pointers, Representation of Polynomials, Representation of sparse matrix in triplet form, Data Structures: Classifications (Primitive & Non-Primitive), Data structure Operations, Stacks: Stacks, Stacks Using Dynamic Arrays, Applications of Stacks – Infix to Postfix Conversion and Postfix Expression Evaluation

(Text Book-1: Chapter 1: 1.2 Chapter 2: 2.3, 2.4, 2.5 Chapter 3: 3.1, 3.2, 3.6)

UNIT – II	06 Hours

Queues & Linked Lists I:

Queues: Queues, Circular Queues Using Dynamic Arrays, Priority Queues, Double ended Queues. **Linked Lists:** Singly Linked Lists (SLL), Operations on SLL. **Primitive Operations:** Insertion and deletion of node at both ends, Display the Linked list, searching for a given node. Representation of polynomials using linked lists.

(Text Book-1: Chapter 3: 3.3, 3.4 Chapter 4)

UNIT – III	06 Hours

Linked List II

Doubly Linked lists (DLL): Basic Operations: Insertion and deletion of node at both the ends, Display the

Linked list, searching for a given node, **Circular Linked List:** Circular SLL and Circular DLL Implementation and primitive insert and delete Operations, **Additional operations on Linked Lists:** Insertion and deletion of nodes at any given position, Searching and deletion of nodes with given value, count nodes, concatenate 2 SLL, display mid element in the list, finding sum and average of list with nodes having integer values. (Text Book-1: Chapter 4)	
UNIT – IV	**06 Hours**
TREES: **Introduction to Trees:** Tree terminologies, Tree classifications, General Tree Representation using DLL nodes, **Binary Trees**: Tree Traversals: Preorder, Inorder, Postorder, **Binary Search Tree**: Creation of BST, insert node into BST, Search BST, examples on Building and Evaluating Binary Expression Trees, **Threaded Binary Trees:** types, representations, and advantages. (Text Book-1: Chapter 5: 5.1, 5.3, 5.5,5.7)	
UNIT – V	**06 Hours**
Heaps, Hashing & Graphs **Heap:** Definition and properties, Implementation of min or max heaps, **Hashing:** Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining, **Graphs:** Disjoint sets, Representation of Graphs - Adjacency/ Cost Matrix, Adjacency Lists. Traversal methods: Breadth First Search / Depth First Search. (Text Book1: Chapter 8: 8.1,8.2,8.3,) (Reference Book 2: Chapter 11)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **Efficient Data Handling & Sustainability:** Use arrays and linked structures for managing large datasets, ensuring memory efficiency and supporting sustainable computing practices.
- **Intelligent Search & Decision Making:** Apply stacks, queues, and trees in AI search algorithms (DFS, A*, game solving) with transparent and explainable logic.
- **Real-Time & Streaming AI Systems:** Utilize linked lists and queues for handling streaming data, real-time updates, and incremental learning in AI pipelines.
- **Graph-Based AI Applications:** Implement graphs for pathfinding, Graph Neural Networks (GNNs), and relationship modeling with fair and ethical data representation.
- **Priority & Resource Optimization:** Use heaps and hashing for task scheduling, fast lookup, and fair prioritization while ensuring privacy and avoiding bias.
- **Responsible & Explainable AI Systems:** Structure data hierarchically (trees) and securely (hashing) to enable interpretable, transparent, and privacy-aware AI solutions.

Applications:

1. Chatbot Conversation Management (Queues & Stacks)
2. Search Engines (Graphs & Trees)
3. Recommendation Systems (Heaps & Hashing)
4. Pathfinding in Robotics / Games (Graphs)
5. Autocomplete & Spell Check (Trees)
6. Memory Management in AI Systems (Linked Lists)
7. Priority-Based Task Scheduling (Heaps)
8. Laboratory Component

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

1. Implement Stack using dynamically allocated arrays.
2. Implement applications of structures.
3. Implement different Queue data structures.
4. Implement basic operations on Singly Linked List.
5. Represent Polynomial using Linked Lists.
6. Implement DLL and Circular Linked Lists.
7. Perform advanced operations on linked lists.
8. Implement Binary Search Tree.
9. Implement Heap and Hash Table.
10. Implement Graph using different representations.

Open Ended programs

1. Design a menu-driven application using Linked Lists to manage student records.
2. Develop a calculator that performs
    - Infix to Postfix
    - Postfix Evaluation
    - Parentheses Matching
    - Prefix Conversion (Bonus)
3. Develop a patient management system using Priority Queue.
4. Develop a Library Management System.
5. Design a graph-based application representing cities connected through roads.

Mini Project Component

- **Industry-Driven Projects (Google / Amazon Style)**

Some sample projects described below simulate the **real software engineering problems**. **Project 1 – Search Engine Indexing**

- **Problem:** Design a data structure that efficiently stores and retrieves search queries.

- **Concepts used:** Hash tables, trees, indexing algorithms
- **Industry relevance:** Search engines (Google).

Project 2 – Social Network Graph Analyzer

- **Problem:** Build a system to analyze connections in a social network.
- **Concepts used:** Graph algorithms, BFS / DFS, Shortest path
- **Industry relevance:** Social platforms (Meta, LinkedIn).

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Understand and apply pointers, dynamic memory allocation, and structures to represent and manage data efficiently.	L2
CO2	Implement stack and queue data structures and use them to solve basic computational problems.	L3
CO3	Develop and manipulate linked list structures to perform various data operations dynamically.	L3
CO4	Apply tree data structures and traversal techniques to organize and process hierarchical data.	L3
CO5	Apply advanced data structures like heaps, hashing, and graphs to solve complex real-world problems.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2		3							3	2
CO2	3	3	2	2	3							3	2
CO3	3	3	2	2	3							3	2
CO4	3	3	2	2	3							3	2
CO5	3	3	3	2	3						2	3	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Ellis Horowitz, Susan Anderson-Freed, and Sartaj Sahni, "Fundamentals of Data Structures in C", 2nd Edition, Orient Longman, 2008.
2. A.M. Tannenbaum, Y Langsam, M J Augentien "Data Structures using C", 1st Edition, Pearson, 2019.

REFERENCE BOOKS:

1. Data Structures using C and C++, Yedidyah Langsam Moshe J. Augenstein and Aaron M. Tenenbaum, 2nd Edition, Pearson, 2009
2. A.M Padma Reddy," Approach of Data Structures", 5th Edition Person Publication,

2015

3. Richard F. Gilberg and Behrouz A. Forouzan: Data Structures A Pseudocode Approach with C

4. Data Structures Using C, Reema Thareja, 1st Edition, 2011, Oxford Higher Education, ISBN-13: 978-0198099307.

E-RESOURCES:

1. https://nptel.ac.in/courses/106102064

2. https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms

3. https://www.udemy.com/topic/data-structures/free/

4. https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures

5. https://cse01-iiith.vlabs.ac.in/

6. https://kremlin.cc/k&r.pdf

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion. E.g.,
2. Role play E.g., Stack, Queue, etc.,
3. Demonstration of solution to a problem through programming.
4. Flip class activity Eg. Arrays, pointers, dynamic memory allocation, etc..

****

Sensors, Systems & IoT Fundamentals

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	04
Hours / Week	:	04 Hours	**Total Hours**	:	60 Hrs (45Hrs L +15Hrs P)
L–T–P-J	:	3–0–2-0			

Course Vision:

The course aims to develop a strong foundation in sensors, embedded systems, and Internet of Things (IoT) technologies for designing intelligent and connected systems. It enables students to understand IoT architectures, communication protocols, and data-driven decision-making for real-world applications. The course emphasizes integration of sensing, networking, cloud computing, and security concepts in smart systems development. It also prepares students for emerging technologies such as edge computing, AI-enabled IoT, industrial automation, and smart infrastructure solutions.

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamental concepts of IoT architecture, M2M communication, and Industry systems.
2. To analyze sensor technologies along with measurement and interfacing techniques.
3. To design and implement embedded systems for IoT applications using modern hardware platforms.
4. To evaluate IoT communication protocols and networking technologies for efficient data transmission.
5. To apply cloud computing, data analytics, and security principles in IoT-based solutions.

Broad Course Description

This course provides a comprehensive foundation in Sensors, Systems, and Internet of Things (IoT), covering the complete IoT stack from sensing and embedded systems to communication protocols and cloud-based analytics. Students will learn the principles of sensor technologies, embedded system design, and IoT networking, along with modern advancements such as edge computing, AI integration, and IoT security. The course emphasizes real-world engineering Applications including smart cities, industrial automation, healthcare, and sustainable systems, while fostering critical thinking through AI-assisted learning and research-integrated pedagogy.

UNIT – I	08 Hours

INTRODUCTION TO IoT

Introduction, Physical design of IoT, Logical design of IoT, IoT Enabling Technologies, IoT Levels & Deployment Templates.

IoT and M2M, Difference between IoT and M2M, SDN and NFV for IoT, software defined networking, and network function virtualization.

(Text Book-1: Chapter 1 and 3)

UNIT–II	09 Hours

SENSOR AND SYSTEMS:

Sensors, signals and systems, Sensors Classification, Sensor Characteristics: Static and dynamic characteristics (accuracy, precision, calibration error, hysteresis, non-linearity, Saturation, Repeatability, Dead Band and Resolution).

Physical Principles of Sensing, Pressure sensors: Mercury pressure sensor, Piezoresistive sensors and Vacuum sensors, Humidity and Moisture Sensors and Temperature Sensors: RTD, Thermoelectric Sensors and Optical Temperature sensors.

(Text Book-2: Chapter 1,3,4,11,14 and 17)

UNIT III	10 Hours

IoT Physical Devices and Cloud Storage Models:

Basic building blocks of an IoT device, Raspberry Pi board, Raspberry Pi interfaces-serial, SPI, I2C, Programming Raspberry Pi with Python-controlling LED with Raspberry Pi, Interfacing an LED and switch with Raspberry Pi, Interfacing a light sensor (LDR) with Raspberry Pi, other IoT Devices, loT Physical

Servers & Cloud Offerings: Introduction to Cloud Storage Models & Communication APIs WAMP-AutoBahn for loT, Xively Cloud for loT. (Text Book-1: Chapter 7 and 8.1,8.2,8.3)	
UNIT – IV	**09 Hours**
IoT CONNECTIVITY AND COMMUNICATION PROTOCOLS: Networking fundamentals for IoT, Layered architecture (TCP/IP stack), Short-range communication (Bluetooth, Zigbee), Long-range communication (LoRa, NB-IoT), Application layer protocols (MQTT, CoAP, HTTP), Addressing and IPv6 (6LoWPAN), QoS and reliability issues in IoT networks. (Text Book-3: Chapter 4, 5 and 8)	
UNIT – V	**09 Hours**
DATA ANALYTICS for IoT: Introduction, Apache Hadoop, MapReduce Programming Model, Hadoop MapReduce Job Execution MapReduce Job Execution Workflow Hadoop Cluster Setup, Using Hadoop MapReduce for Batch Data Analysis Hadoop YARN, REST based approach and WebSocket based approach. (Text Book-1: Chapter 10)	

Lab Experiments:

1. To Interface LED with Raspberry Pi and write a program to turn ON
2. To interface IR/LDR with Raspberry Pi and write a program to turn ON LED at sensor detection.
3. To interface dc motor with Raspberry Pi to control the speed of the motor
4. To interface OLED with Raspberry Pi and write a program to print temperature and humidity readings on it.
5. To interface Bluetooth with Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.
6. Write a program on Raspberry Pi to upload temperature and humidity data to thingspeak cloud.
7. Write a program on Raspberry Pi to retrieve temperature and humidity data from thingspeak cloud.
8. Write a program on Raspberry Pi to publish temperature data to MQTT broker.
9. Write a program to create TCP server on Raspberry Pi and respond with humidity data to TCP client when requested.
10. Write a program to create UDP server on Raspberry Pi and respond with humidity data to UDP client when requested.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-assisted IoT architecture design and intelligent M2M communication analysis.
- AI-based sensor data processing, calibration, and fault detection.
- AI-assisted embedded system programming and TinyML-based edge intelligence.
- AI-driven network optimization and adaptive IoT communication protocols.
- AI-based data analytics, predictive modeling, and intelligent IoT security systems.
- Students evaluate limitations and ethical concerns of AI-designed IoT systems and M2M architectures.
- Students analyse reliability and bias in AI-processed sensor data and automated actuation.
- Students assess constraints of AI-generated embedded solutions in real-time

environments.
  - Students examine risks, privacy issues, and trust in AI-managed communication networks.
  - Students evaluate trade-offs between data intelligence, privacy, and reliability in AI-driven IoT systems.

Applications:

1. Smart Home Automation Systems (Sensors, Embedded Systems, IoT Communication)
2. Environmental Monitoring Systems (Air Quality, Temperature, Pollution Sensors)
3. Industrial Automation & Predictive Maintenance (IIoT, Sensor Networks, Data Analytics)
4. Smart Healthcare & Wearable Devices (Biosensors, Real-time Monitoring, Cloud Integration)
5. Intelligent Transportation Systems (Vehicle Tracking, Traffic Monitoring, IoT Networks)
6. Smart Agriculture Systems (Soil Moisture Sensors, Automated Irrigation, IoT Control)
7. Energy Management & Smart Grids (Smart Meters, Energy Optimization, IoT Analytics)

Mini Project Component:

Some sample projects described below simulate the real software engineering problems.

Project 1 – Smart Environmental Monitoring System.

- Problem: Design an IoT system to monitor temperature, humidity, and air quality in real-time.
- Concepts used: Sensors (temperature, gas), embedded systems (ESP32/Arduino), cloud integration, data visualization
- Industry relevance: Smart cities, environmental monitoring agencies, pollution control systems.

Project 2 – Industrial Predictive Maintenance System

- Problem: Develop an IoT-based system to monitor machine health and predict failures.
- Concepts used: Vibration/temperature sensors, data analytics, cloud platforms, AI-based anomaly detection.
- Industry relevance: Manufacturing industries, Industry 4.0, companies like Siemens, GE.

Project 3 – Smart Home Automation System

- Problem: Build an IoT system to control home appliances remotely and automatically.
- Concepts used: Embedded systems, IoT communication protocols (Wi-Fi, MQTT), mobile/cloud interface.
- Industry relevance: Smart home companies (Google Nest, Amazon Alexa ecosystem).

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain the architecture, physical & logical design of IoT systems, IoT levels, M2M differences, and SDN/NFV concepts	L3
CO2	Analyze the static and dynamic characteristics of measurement systems and select appropriate sensors for IoT Applications	L4
CO3	Design and implement IoT physical device solutions using Raspberry Pi with Python for interfacing sensors, actuators, and communication peripherals	L6
CO4	Compare and apply IoT communication protocols (BLE, Zigbee, LoRa, MQTT, CoAP, 6LoWPAN) for appropriate network design scenarios	L3
CO5	Evaluate data analytics pipelines and security mechanisms for secure IoT deployments.	L5

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PSO 1	PSO 2
CO1	3	2	2	2	2				2		2	2	2
CO2	3	2	2	2	2				2		2	2	2
CO3	3	2	2	2	2				2		2	2	2
CO4	3	2	2	2	2				2		2	2	2
CO5	3	2	2	3	2				2		2	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Internet of Things: A Hands-On Approach — Arshdeep Bahga & Vijay Madisetti (Universities Press / VPT, 2015).
2. Handbook of Modern Sensors: Physics, Designs, and Applications" — Jacob Fraden (5th Ed., Springer, 2016).
3. Fundamentals of IoT Communication Technologies" — Rolando Herrero (Springer, 2021).
4. "Cloud Computing: A Hands-On Approach" — Arshdeep Bahga & Vijay Madisetti (companion volume, 2014).

REFERENCE BOOKS:

1. IoT Security: Advances in Authentication" — Madhusanka Liyanage et al. (Wiley, 2020)
2. Vinod Kumar Khanna, IoT Sensors: An Exploration of Sensors for Internet of Things, CRC Press / Taylor & Francis, 2024.
3. Lei Jing, Yoshinori Matsumoto, Zhan Zhang (Eds.), Exploring IoT Sensors and Their Applications: Advancements, Challenges, and Opportunities in Smart Environments, MDPI, 2024.
4. Alexandru Lavric et al. (Eds.), IoT Multi-Sensors, MDPI, 2023.
5. Behnam Mobaraki et al. (Eds.), Advances in Intelligent Sensors and IoT Solutions, MDPI, 2024.

E-RESOURCES:

1. NPTEL (IoT, Embedded Systems, Sensor courses by IITs)
2. MIT OpenCourseWare (Embedded Systems, IoT fundamentals)
3. IEEE research articles on IoT, AIoT, and sensor networks

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion. E.g., Designing a smart home or smart agriculture IoT system.
2. Role play. E.g., Students act as sensors, gateways, and cloud components to simulate IoT data flow.
3. Demonstration of solution to a problem through programming. E.g., Interfacing a sensor with Arduino/ESP32 and sending data to cloud/dashboard.
4. Flip class activity. E.g., Communication protocols (MQTT, CoAP), cloud platforms, and IoT security concepts.

****

Sustainability Engineering & ESG

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs L
L–T–P-J	:	2–0–0-0			

Course Vision: This course aims to develop a strong understanding of sustainability principles and ESG (Environmental, Social, and Governance) frameworks in the context of digital and engineering systems. It aims to enable students to evaluate the environmental and social impact of modern technologies and design responsible, resource-efficient solutions. The course further aims to prepare learners to use data-driven and AI-based approaches for sustainability analysis, supporting informed decision-making toward environmentally responsible and socially inclusive technology development.

Course Learning Objectives:

This Course will enable students to:

1. Understand sustainability principles and ESG frameworks
2. Analyze environmental impacts of digital and engineering systems
3. Apply lifecycle and carbon footprint analysis using data tools
4. Evaluate social and governance aspects in technology systems
5. Use AI tools for sustainability analytics and decision-making

Broad Course Description

The course **Sustainability Engineering and ESG Analytics for Digital Systems** provides a comprehensive understanding of sustainability principles and ESG (Environmental, Social, and Governance) frameworks with a focus on computing and technology-driven domains. It covers fundamental concepts such as environmental impact assessment of digital systems, resource optimization in cloud computing, and sustainability challenges in smart cities and digital platforms. The course further explores analytical approaches including lifecycle assessment and carbon footprint evaluation of IT infrastructure, along with social and governance considerations in technology systems. It also introduces the use of AI and data-driven techniques for sustainability analysis, prediction, and decision-making. Overall, it equips students with the skills to evaluate, design, and optimize technology solutions that are environmentally responsible, socially inclusive, and aligned with ESG standards.

UNIT – I	06 Hours

SUSTAINABILITY FUNDAMENTALS & ESG:

Topics:

Sustainable development, Triple Bottom Line, ESG framework, SDGs, ESG vs CSR, sustainability challenges in digital economy.

Engineering Applications:

Green computing, sustainable IT infrastructure, energy-efficient data centers.

AI Integration:

AI-based sustainability trend analysis, ESG scoring tools.

Authentic Intelligence:: Reflection:

Students critically analyze AI-generated sustainability insights and validate them using real-world data.

SDG Goals: SDG 4 (Education), SDG 9 (Innovation)

UNIT – II	06 Hours

Environmental Sustainability in Digital Systems (6 Hours): Topics:

Carbon footprint of IT systems, energy consumption in data centers, lifecycle assessment (LCA), e-waste management, green cloud computing. Engineering Applications: Energy-efficient servers, cloud optimization, sustainable software engineering.

AI Integration: AI models for energy prediction and carbon footprint estimation.

Authentic Intelligence: Reflection: Students compare AI predictions with real energy consumption data.

SDG Goals: SDG 7, SDG 13	
UNIT – III	**06 Hours**
Social Sustainability & Ethical Computing (6 Hours): Syllabus Topics: Stakeholder analysis, digital inclusivity, data privacy, ethical AI, cybersecurity and society, social impact of automation. Engineering Applications: AI ethics in recommendation systems, fairness in algorithms, inclusive software design. **AI Integration:** Sentiment analysis, bias detection in AI systems. **Authentic Intelligence:** Reflection Students analyze bias and fairness in AI-generated outputs. SDG Goals: SDG 10, SDG 16	
UNIT – IV	**06 Hours**
Governance, ESG Reporting, Sustainable Design & Future Technologies (6 Hours): **Syllabus Topics:** Corporate governance, ESG reporting frameworks (GRI, TCFD), ESG metrics, sustainability KPIs, data visualization, Circular economy, sustainable software design, green AI, smart cities, Industry 4.0 sustainability, digital twins for sustainability Engineering Applications: Corporate ESG dashboards, compliance analytics, risk management systems, Smart grids, intelligent transportation, sustainable IoT systems. **AI Integration:** AI-based ESG report analysis, dashboards using Power BI/Python, AI-based optimization of sustainable systems. **Authentic Intelligence:** Reflection: Students evaluate transparency and reliability of ESG reports. Students assess trade-offs between performance and sustainability SDG Goals: SDG 11, SDG 12, SDG 16, SDG 17	
UNIT – V	**06 Hours**
Energy Efficiency, Audit & Smart Energy Systems (6 Hours) Energy efficiency: Definition, importance, and metrics (energy intensity, efficiency ratio), Energy losses in computing and electronic systems (processors, data centers, networks), Energy-efficient computing: Green computing principles, low-power design, virtualization, Energy Audit: Definition, types (preliminary and detailed audit), steps involved, Energy audit of IT infrastructure: servers, cooling systems, networks, Energy-saving techniques in data centers and embedded systems , Introduction to smart energy management systems Engineering Applications: • Energy-efficient data centers and cloud computing systems • Power optimization in embedded systems and IoT devices • Energy audit of IT infrastructure in industries and campuses • Smart energy monitoring in communication networks	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-based energy consumption prediction using machine learning models, Optimization of energy usage in data centers using AI algorithms, Smart energy monitoring using IoT and AI analytics, Data-driven identification of energy loss points
- Students will critically evaluate AI-based energy optimization models by comparing predicted energy savings with actual system performance. They will analyse the trade-offs between computational performance and energy efficiency, and assess the reliability and scalability of AI-driven energy management systems.

Teaching-Learning Process (AI-Enhanced Pedagogy)

Lecture Method: Conceptual teaching supported by AI-generated visualizations

- Interactive Learning: AI-assisted problem solving and discussion
- Video/Simulation: Use of dashboards, sustainability simulators
- Collaborative Learning: Group ESG case studies using datasets
- Higher Order Thinking: AI output validation and critical evaluation
- Problem-Based Learning: Real-world sustainability problems in IT systems
- Multiple Solution Approaches: Analytical + AI-based evaluation
- Real-World Applications: Cloud sustainability, green computing, ESG analytics
- Flipped Learning: ESG problems will be given and Students are asked to solve using AI tools

Applications:

1. Energy-efficient data centers and cloud computing systems
2. Power optimization in embedded systems and IoT devices
3. Energy audit of IT infrastructure in industries and campuses
4. Smart energy monitoring in communication networks

Mini Project Component

Requirements:

- Teams of 3–4 students
- Use AI/data tools
- Real-world ESG problem

Evaluation:

- Sustainability understanding – 40%
- AI implementation – 30%
- Presentation – 20%
- Report – 10%

Suggested Projects

- Carbon footprint analysis of cloud platforms
- ESG dashboard for a company
- AI model for energy consumption prediction
- Sustainable smart city data analytics
- Bias detection in AI systems

CO	Course Outcome	Bloom's Level	AI Linkage
CO1	Understand sustainability principles and ESG frameworks	L2	AI-based trend analysis
CO2	Apply lifecycle and carbon footprint analysis	L3	OpenLCA, Python
CO3	Analyze environmental and social impacts in digital systems	L3	Data analytics tools
CO4	Evaluate ESG performance using metrics and reports	L4	Power BI, ESG datasets
CO5	Assess ethical and governance issues in technology systems	L4	AI-based risk analysis

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			2		3					2	
CO2	3	3	2	2	3		3					3	
CO3	3	3	2	2	3	2	3	2				3	
CO4	3	3	2	3	3		3			2		3	
CO5	2	2		2	2	3	3	3	2	2		2	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

- Allen & Shonnard – Sustainable Engineering
- ESG and Sustainable Finance – Reference Text

REFERENCE BOOKS:

1. UN SDG Reports
2. World Bank ESG ReportsSustainability journals (Elsevier, IEEE)

Additional AI & Computational Resources:

- Python (Collab notebooks)
- ESG datasets (World Bank, UN)
- NPTEL Sustainability Courses
- MIT Open Courseware

****

AI Powered – Full Stack Web Development

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	02
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs (15Hrs L +15Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: This course aims to equip students with comprehensive skills in full stack web development integrated with modern AI capabilities, enabling them to design, develop, and deploy intelligent, production-ready web Applications. It aims to build proficiency in both frontend and backend technologies along with API-driven architectures, cloud deployment, and database management. The course further aims to prepare learners to create scalable, secure, and AI-enhanced digital solutions that align with current industry practices in software engineering and intelligent systems development.

Course Learning Objectives:

This Course will enable students to:

1. Build proficiency in modern frontend frameworks (React.js) and backend development for full stack Applications.
2. Integrate AI and ML capabilities into web Applications using REST APIs and language model libraries.
3. Understand RESTful API design, microservices architecture, and cloud deployment strategies.
4. Develop production-ready, AI-enhanced web Applications with authentication and database integration

Broad Course Description

AI-Powered Full Stack Web Development bridges traditional web development with transformative AI capabilities. The course covers the complete development pipeline: React.js frontends with hooks and state management, Node.js and Python FastAPI backends, JWT/OAuth 2.0 authentication, MongoDB and PostgreSQL integration, AI API integration (OpenAI, Gemini, Hugging Face), Retrieval-Augmented Generation (RAG) with vector databases, Docker containerization, CI/CD pipelines with GitHub Actions, and cloud deployment (Vercel, Railway, AWS). Emphasis is on building scalable, secure, AI-enhanced production Applications that demonstrate full-pipeline competency from design to deployment.

UNIT – I	03 Hours

Frontend Development with React.js: Modern HTML5/CSS3, JavaScript ES6+ (Promises, Async/Await, Modules), React.js – Components, Props, State, Hooks (useState, useEffect, useContext), React Router for SPA, Redux Toolkit/Zustand state management, Tailwind CSS, AI-assisted UI generation tools. (Ref: Robin Wieruch, The Road to React Ch. 1–5)

Textbook Reference:

• Robin Wieruch, The Road to React – Ch. 1–5 (Self-published, 2023)

• Alex Banks & Eve Porcello, Learning React, 2nd Ed – Ch. 1–4 (O'Reilly, 2020)

UNIT – II	03 Hours

Backend Development with Node.js & Python: Node.js/Express REST API design principles, Middleware, JWT/OAuth 2.0 authentication, Python FastAPI/Flask for AI backends, MongoDB (Mongoose), PostgreSQL (Prisma ORM), WebSockets for real-time communication, API rate limiting and security, Cloud storage (AWS S3 / Cloudinary). (Ref: Ethan Brown, Web Development with Node and Express Ch. 1–7)

Textbook Reference:

• Ethan Brown, Web Development with Node and Express – Ch. 1–7 (O'Reilly, 2019)

• FastAPI Official Documentation (fastapi.tiangolo.com)

UNIT – III	03 Hours

AI Integration in Web Applications: OpenAI GPT, Google Gemini, Anthropic Claude APIs, Building chatbot interfaces, Prompt engineering for web Applications, Hugging Face models for NLP and image tasks, Retrieval-Augmented Generation (RAG) – embeddings, vector databases (Pinecone, ChromaDB), AI-powered search and recommendations, Streaming AI responses in real time. (Ref: NVIDIA DLI Building RAG Agents With LLMs)

Textbook Reference:

• NVIDIA DLI: Building RAG Agents With LLMs (Course Material, learn.nvidia.com)
• OpenAI API Reference (platform.openai.com/docs)

UNIT – IV	03 Hours

Full Stack Architecture & Security: Monolithic vs. Microservices architecture, API Gateway and service communication, Role-based access control (RBAC), Input validation and sanitization, CORS, HTTPS, CSP headers, Secrets management with environment variables, Error handling and logging, Redis caching, Performance optimization – lazy loading, code splitting. (Ref: Sam Newman, Building Microservices Ch. 1–5)

Textbook Reference:

• Sam Newman, Building Microservices, 2nd Ed – Ch. 1–5 (O'Reilly, 2021)
• OWASP Top 10 Web Application Security Risks (owasp.org)

UNIT – V	03 Hours

DevOps, Testing & Deployment: Git/GitHub branching strategies, Unit/Integration/E2E testing with Jest/Cypress, Docker and Docker Compose containerization, CI/CD with GitHub Actions – workflows and triggers, Cloud deployment (Vercel, Railway, AWS Lambda basics), Monitoring with Sentry and Vercel Analytics, Introduction to serverless architecture. (Ref: Jez Humble, Continuous Delivery Ch. 1–4)

Textbook Reference:

Jez Humble & David Farley, Continuous Delivery – Ch. 1–4 (Addison-Wesley, 2010)

Docker Official Documentation (docs.docker.com)

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Unit 3	**Generative AI Explained**	Generative AI	1 Hr	—	courses.nvidia.com/course-v1:DLI+S-FX-07+V1	Beginner-level conceptual foundation for how GenAI works — essential prerequisite before students integrate OpenAI, Gemini, or Claude APIs in Unit 3 chatbot and AI feature development.
Unit 3	**Augment Your LLM Using RAG**	Generative AI / LLM	2 Hrs	—	courses.nvidia.com/course-v1:NVIDIA+S-FX-16+v1	Introduces RAG pipeline fundamentals and NVIDIA AI Foundations endpoints — builds the conceptual and practical base needed before students implement vector-DB-backed document Q&A in Unit 3.

Unit 3	**Building RAG Agents with LLMs**	Generative AI / LLM	8 Hrs	**Yes**	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+C-FX-15+V1	Core 8-hr free course on end-to-end LLM systems with LangChain, LangServe, FAISS, Gradio — directly maps to RAG chatbot and AI search implementation in Unit 3. Certificate awarded on completion.
Unit 3	**Techniques for Improving RAG Systems**	Generative AI / LLM	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-FX-15+V1	Advanced RAG optimization techniques (re-ranking, hybrid search, contextual compression) — extends Unit 3's AI integration module towards production-quality retrieval systems.
Units 3–5	**Deploying RAG Pipelines at Scale**	Generative AI / LLM	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-FX-16+V1	Deploying RAG pipelines at scale using Triton Inference Server — bridges Unit 3 AI integration with Unit 5 cloud deployment, teaching scalable AI model serving.
Unit 3	**Intro to Transformer-Based NLP**	Generative AI / NLP	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+S-FX-08+V1	Transformer and BERT/GPT fundamentals for text classification, NER, QA tasks — directly underpins Hugging Face NLP model integration and text-based AI features in Unit 3.
Unit 3	**Prompt Engineering With Llama 2**	Generative AI / LLM	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-FX-12+V1	Prompt engineering with zero-shot, few-shot, and chain-of-thought strategies — directly applicable to designing effective prompts for OpenAI/Gemini API calls in Unit 3 AI integration module.
Units 4–5	**AI Infrastructure and Operations Fundamentals**	Infrastructure / AI Ops	8 Hrs	**Yes**	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+S-IN-01+V1	GPU cluster management, NVIDIA AI Enterprise, container orchestration — supports cloud deployment, containerized AI serving, and monitoring topics in Units 4 and 5.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI as Co-Developer:** Integrate GitHub Copilot and ChatGPT into the development workflow; evaluate code quality, security, and maintainability.

- **Reflective Prompt Engineering:** Iteratively refine prompts and critically evaluate AI output quality, hallucinations, and edge-case failures.
- **AI Ethics in Web Products:** Discuss bias, transparency, and user privacy in AI-powered web features and recommendation systems.
- **Comparative Analysis:** Compare hand-coded solutions with AI-generated code for quality, readability, performance, and security implications.
- **Human-in-the-Loop Systems:** Design web features where AI assists but humans retain final decision authority and audit capabilities.

Applications:

- AI-Powered E-Commerce Platforms with Recommendation Engines
- Healthcare Telehealth Portals with AI Diagnostic Support
- EdTech Platforms with Personalized AI Learning Paths
- Enterprise SaaS Applications with AI-Driven Analytics
- Social Platforms with AI-Powered Content Moderation

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Build a responsive React.js SPA with dynamic routing, custom hooks, and Tailwind CSS.
- Create a RESTful API with Node.js/Express: CRUD operations, JWT authentication, and MongoDB.
- Integrate OpenAI/Gemini API to build a conversational chatbot interface in a web application.
- Implement a RAG pipeline: document loading, embeddings, vector DB storage, and web-based Q&A.
- Build a real-time collaborative feature using WebSockets (live chat or collaborative editor).
- Develop an image recognition web feature using Hugging Face inference API in a React frontend.
- Design role-based authentication with JWT, refresh tokens, and OAuth social login (GitHub/Google)
- Create an AI-powered semantic search engine using text embeddings and cosine similarity.
- Containerize a full stack application using Dockerfile and Docker Compose.
- Deploy a full stack AI app via CI/CD pipeline with GitHub Actions to Vercel and Railway.

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – AI Study Assistant Platform

Problem: Students upload study material (PDF/text) and ask questions using a RAG-based AI backend.

Concepts used: React, FastAPI, RAG pipeline, Vector DB, OpenAI API

Industry relevance: EdTech and corporate training platforms.

Project 2 – Smart Job Board with AI Matching

Problem: AI matches candidates to job roles using skill profile embeddings and job description similarity.

Concepts used: Full stack, NLP embeddings, Semantic search, Recommendation

Industry relevance: LinkedIn and Indeed-style recruitment platforms.

Project 3 – AI Health Symptom Checker

Problem: Users input symptoms; AI provides guided health information with doctor referral integration.

Concepts used: React, Node.js, LLM APIs, Authentication, Database

Industry relevance: Healthcare IT and telemedicine platforms.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Design and build interactive React.js frontend Applications with modern state management and AI-assisted UI tools.	L3
2	Develop secure, scalable RESTful backend services using Node.js/Python with database integration and authentication.	L3
3	Integrate AI APIs and machine learning models into web Applications to create intelligent, context-aware features.	L3
4	Architect full stack Applications following security best practices, microservices patterns, and performance optimization.	L4
5	Deploy AI-powered web Applications using containerization (Docker), CI/CD pipelines, and cloud platforms.	L5

Table: Mapping Levels of COs to POs / PSOs													
	Program Outcomes (POs)											**Program Specific Outcomes (PSOs)**	
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	3	3	3						2	3	3
CO2	3	3	2	3	3						2	3	2
CO3	3	3	3	3	3						3	3	3
CO4	3	3	2	3	3						2	3	2
CO5	2	3	3	3	3						2	3	3

3: Substantial (High)	**2: Moderate (Medium)**	**1: Poor (Low)**

Suggested Tools:

- React.js, Next.js – Frontend
- Node.js, Express.js, FastAPI – Backend
- MongoDB, PostgreSQL – Databases
- OpenAI, Gemini, Hugging Face – AI APIs
- Docker, GitHub Actions – DevOps
- Vercel, Railway – Cloud Deployment

TEXT BOOKS:

1. Robin Wieruch, The Road to React, Self-published, 2023.
2. Ethan Brown, Web Development with Node and Express, O'Reilly, 2019.
3. Sam Newman, Building Microservices, 2nd Ed, O'Reilly, 2021.

REFERENCE BOOKS:

1. Alex Banks & Eve Porcello, Learning React, 2nd Ed, O'Reilly, 2020.
2. OWASP Top 10 – Web Application Security Risks (owasp.org).

E-RESOURCES:

- https://theodinproject.com
- https://www.freecodecamp.org/learn
- https://platform.openai.com/docs
- https://learn.nvidia.com (NVIDIA DLI Free Courses)

Activity Based Learning (Suggested Activities in Class)

- Hackathon – build and deploy an AI-integrated web app in 24 hours.
- Code review sessions – peer evaluation of GitHub repositories using structured rubrics.
- Guest lectures from industry practitioners on full stack architecture and AI product development.
- Demo Day – each team presents their deployed AI web application to simulated industry judges.

****

GitHub & Version Control

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	02
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs (15Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: This course aims to build strong foundational skills in version control systems using Git and collaborative development using GitHub. It aims to enable students to efficiently manage source code, track changes, and work effectively in team-based software projects. The course also aims to prepare learners for modern industry workflows by developing proficiency in branching, merging, and distributed development practices essential for scalable and professional software engineering.

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamentals of version control systems.
2. Learn Git commands and workflows.
3. Use GitHub for collaboration and project management.
4. Apply branching, merging, and team workflows in real projects.

Broad Course Description

Version control is a fundamental concept in software development that enables individuals and teams to track, manage, and organize changes made to files over time. A version control system (VCS) maintains a history of modifications, allowing developers to revert to previous versions, compare changes, and collaborate efficiently without overwriting each other's work. Modern systems like Git follow a distributed model, where every user has a complete copy of the repository, ensuring reliability, flexibility, and offline access. Key features include branching (working on separate features independently), merging (combining changes), and maintaining a structured development workflow.

GitHub is a cloud-based platform built around Git that enhances version control with powerful collaboration tools. It allows developers to host repositories online, share code, and work together through features like pull requests, issue tracking, and code reviews. GitHub plays a crucial role in open-source development and enterprise projects by enabling transparency, teamwork, and continuous integration practices. Together, Git and GitHub form the backbone of modern software engineering, supporting scalable development, efficient collaboration, and robust project management across diverse domains.

UNIT – I	03 Hours

Introduction to Version Control:

Concept of Version Control Systems (VCS), Types: Centralized vs Distributed VCS, Introduction to Git, Installation and setup, Basic Git workflow

Textbook 1: Chapter: 1, 2

Textbook 2: Chapter: 1, 2, 3

UNIT – II	03 Hours

Basic Git Operations:

Repository creation (local), Staging, committing, viewing history, Working directory, staging area, repository, Undoing changes

Textbook 1: Chapter 2

Textbook 2: Chapter: 4, 5, 6

UNIT – III	03 Hours

Branching and Merging:

Branch creation and switching, Merging branches, Merge conflicts and resolution, Introduction to workflows.

Textbook 1: Chapter: 3

Textbook 2: Chapter: 11, 13, 14

UNIT – IV	03 Hours

Remote Repositories & GitHub: Creating repositories on GitHub, Cloning, pushing, pulling, Forking and pull requests, Issues and basic project management Textbook 1: Chapter: 2, 6 Textbook 2: Chapter: 7, 8, 9, 10, 12	
UNIT – V	**03 Hours**
Collaboration & Best Practices: Team workflows (Git Flow basics), Code review process, README, documentation, Introduction to CI/CD concepts Textbook 1: Chapter: 5, 6 Textbook 2: Chapter: 16, 17, 19	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Code Versioning:** Students use AI tools (e.g., GitHub Copilot) to generate code and track how versions evolve using Git..
- **Intelligent Commit Message Generation:** AI helps generate meaningful commit messages, while students refine them to ensure clarity and context.
- **AI-Driven Debugging with Version History:** Students use AI to identify bugs and compare commits to understand how issues were introduced and resolved.
- **Ethical Use of AI in Code Repositories:** Students reflect on plagiarism, proper attribution, and responsible AI usage in shared repositories.
- **Integration with CI/CD and AI Testing:** AI-based testing tools integrated with GitHub workflows help automate validation, while students interpret results.

Applications:

1. Software Development Management
2. Team Collaboration
3. Open Source Contribution
4. Project Version Tracking
5. Continuous Integration & Deployment

Laboratory Component

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

1. Install Git and configure user profile
2. Create and manage repositories
3. Perform commits and track changes
4. Work with branches and resolve conflicts
5. Push/pull from GitHub
6. Create pull requests and collaborate

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the **real software engineering problems**.

Project 1 – Collaborative Code Management using GitHub

- **Problem:** Difficulty in managing multiple developers' code changes, version conflicts, and maintaining project history efficiently.
- **Concepts used:** Git basics, branching & merging, pull requests, conflict resolution, repository management..
- **Industry relevance:** Widely used in software companies for team collaboration, DevOps workflows, code review, and continuous integration.

Project 2 – Open Source Contribution Workflow using GitHub

- **Problem:** Beginners struggle to understand how to contribute to open-source projects effectively.
- **Concepts used:** Forking, cloning, branching, pull requests, issue tracking using Git.
- **Industry relevance:** Mirrors real-world open-source and enterprise contribution workflows.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Understand the fundamentals of version control and Git environment setup.	L2
CO2	Apply basic Git commands to manage local repositories.	L3
CO3	Implement branching and merging strategies in Git.	L3
CO4	Use GitHub for remote repository management and collaboration.	L3
CO5	Apply collaborative workflows and best practices using Git and GitHub.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			2						2	3	
CO2	2	2	2		3						2	3	
CO3	2	3	2		3						2	3	
CO4	2	2	2		3						2	3	
CO5	2	2	2		3						2	3	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Suggested Tools:

- Git

- GitHub
- VS Code / any IDE

TEXT BOOKS:

1. Pro Git - Scott Chacon & Ben Straub, Apress, 2nd Edition.
2. Beginning Git and GitHub, Mariot Tsitoara · Apress, 2020.

REFERENCE BOOKS:

1. Version Control with Git – Jon Loeliger & Matthew McCullough, O'Reilly.
2. Git Pocket Guide – Richard E. Silverman, O'Reilly.

E-RESOURCES:

1. https://docs.github.com/en
2. https://git-scm.com/docs

Activity Based Learning (Suggested Activities in Class)

1. Student seminars (on topics of the syllabus and related aspects (individual activity).
2. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams).
3. Study projects (by very small groups of students on selected local real-time problems about the syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity).

****

Indian Knowledge Systems

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs l
L–T–P-J	:	2–0–0-0			

Course Vision: This course aims to develop a strong foundation in language processing and computational thinking by integrating principles of Artificial Intelligence with concepts from the Indian Knowledge System. It aims to enable students to model and solve engineering problems using linguistic, mathematical, and scientific perspectives while appreciating the contributions of Indian mathematics and astronomy. The course also aims to foster interdisciplinary thinking for designing intelligent, culturally grounded, and real-world AI-based solutions.

Prerequisites: Good Understanding of English Language, Python programming language

Course Learning Objectives:

This Course will enable students to:

1. Understand principles of an ecosystem for Language Processing.
2. Model engineering problems computationally using Number Systems and Units of Measurements.
3. Apply AI tools for Knowledge: Framework and Classification.
4. Evaluate unique aspects of Indian Mathematics.
5. Design solutions for real-world engineering problems using Indian Astronomy.

Broad Course Description

Course is designed to develop the ability to model, analyze, and design intelligent engineering systems by integrating foundational Concepts of Science and Technology in Indian Knowledge System to Computing and Linguistics which are required in the Artificial Intelligence era.

UNIT – I	06 Hours

Linguistics: An ecosystem for Sanskrit Language Processing, Components of a language, Paninis's work on Sanskrit grammar, Phonetics in Sanskrit, Four stages of Speech, Patterns in Sanskrit Vocabulary, Computational Concepts in Astadhyayi, Logic for Sentence Construction, Importance of verbs, Role of Sanskrit in Natural Language Processing.

UNIT – II	06 Hours

Number System and Units of Measurement: Number System in India – Historical Evidence, Salient Features of the Indian Numeral System, The concept of Zero and its Importance, Large Numbers and Their Representation, Place value of Numerals, Decimal System, Unique approaches to Represent Numbers, Measures for Time, Distance and Weight, Pingala and Binary system

UNIT – III	04 Hours

Knowledge: Framework and Classification

The knowledge Triangle, Prameya, Pramana, Samshaya, Framework for Establishing Valid Knowledge.

UNIT – IV	06 Hours

Mathematics

Unique aspects of Indian Mathematics, Greate Mathematicians and their contributions, Arithmetic, Geometry, Trigonometry, Algebra, Binary Mathematics and Combinatorial Problems in Chandah sastra of Pingala, Magic Squares in India

UNIT – V	08 Hours

Astronomy Unique aspects of Indian Astronomy, Historical Development of Astronomy in India, The celestial Coordinate system, Elements of the Indian Calander, Aryabhatiya and the Siddhantic Tradition, Panchanga – The Indian Calander System, Astronomical Instruments (Yantras), Jantar Mantar of Raja Jai Singh Sawai

Mini Project Component

The course emphasizes hands-on implementation, practical exploration, and AI-assisted development using Indian Knowledge system.

• Students will design a simple intelligent system using Indian Knowledge System and Applying it on various AI models.

• Examples:

• AI-based Rain prediction system

• Prediction of Solar and Lunar Eclipse based on the astronomical data

• Smart Calanders to manage time efficiently

• Students must analyze:

• System design

• AI modeling

• Reliability

• Ethical implications

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain rule based Grammatical Operations, patterns in vocabulary, stages of speech.	L2
CO2	Model systems to decipher the unique methods deployed to represent numbers	L3
CO3	Apply AI techniques to establish the right knowledge	L3
CO4	Evaluate Mathematical systems developed by Aryabhata and Pingala	L5
CO5	Design intelligent solutions using celestial coordinate system and the basic elements of the Indian Calander	L6

Table: Mapping Levels of COs to POs / PSOs												
COs	Program Outcomes (POs)										Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3						2	2		2	2	
CO2	3	3	2					2	2		2	2	
CO3	3	3	2					2	2		2	2	
CO4	3	3	2					2	2		2	2	
CO5	3	3	2					2	2		2	2	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

6. Introduction to Indian Knowledge System Concepts and Applications, B.Mahadevan, Vinaya Rajat Bhat, Nagendra Pavana R.N., PHI Learning Private Limited, 2024 ISBN-978-93-91818-20-3 Ebook ISBN-978-93-91818-21-0

REFERENCE BOOKS:

1. Kapoor and A. K. Singh, Indian Knowledge Systems (2 Vols. Set). New Delhi, India: D.K. Printworld, 2005.
2. Murthy and K. Venkataraman, Indian Knowledge System: Indian Contributions to Science, Vol. I., 2025. **DOI:** https://doi.org/10.71412/iks0020

E-RESOURCES:

https://onlinecourses.swayam2.ac.in/e-learning/preview/imb24_mg20

Activity Based Learning (Suggested Activities in Class)

Demonstration of solution to a problem through programming.

****

Low Code/No Code Development

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	02
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs (15Hrs L +15Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision:

This course aims to develop practical skills in Low Code/No Code development, enabling students to design, build, and deploy applications rapidly using modern visual development platforms. It aims to empower learners to create scalable digital solutions with minimal traditional coding while understanding workflow automation, API integration, and system design. The course also aims to build critical awareness of the strengths and limitations of LCNC platforms for enterprise applications, fostering innovation, efficiency, and responsible technology use.

Course Learning Objectives:

This Course will enable students to:

1. Understand the Low Code/No Code development paradigm, its ecosystem, and market landscape.
2. Gain practical proficiency in building Applications using leading LCNC platforms like Bubble.io and Webflow.
3. Design and automate multi-step workflows using visual programming and integration tools.
4. Evaluate the limitations, scalability, and best use cases for LCNC platforms in enterprise scenarios.

Broad Course Description

Low Code/No Code (LCNC) Development democratizes application creation, enabling non-technical professionals and accelerating development for engineers. The course covers the LCNC paradigm and platforms including Bubble.io, Webflow, Microsoft Power Platform, Zapier/Make automation, mobile apps (Glide, Adalo, FlutterFlow), Backend-as-a-Service (Airtable, Supabase, Firebase), internal tools (Retool, Appsmith), and AI-powered LCNC platforms that generate functional Applications from natural language prompts. Students learn to design data models, automate complex workflows with branching logic, integrate third-party APIs, and deploy production-ready LCNC Applications with custom domains. The course also critically examines vendor lock-in, scalability ceilings, and responsible AI deployment within the LCNC context.

UNIT – I	03 Hours

Introduction to LCNC Paradigm: Evolution of software development from coding to visual development. LCNC market landscape and leading platform comparison. Citizen developer concept and its business impact. Types of LCNC platforms – process automation, web/app builders, database-driven tools. LCNC vs. Traditional development trade-offs. Introduction to Bubble.io – interface, database, and workflow basics. (Ref: Rui Barbosa, Beginning No-Code Development Ch. 1–3)

Textbook Reference:

• Rui Barbosa & Larry Liu, Beginning No-Code Development – Ch. 1–3 (Apress, 2022)
• Bubble.io Academy Official Documentation (bubble.io/academy)

UNIT – II	03 Hours

Web Application Development with LCNC: Responsive web apps with Bubble.io and Webflow. UI/UX design in visual builders – pages, elements, styling. Database design within LCNC – data types, tables, relationships. User authentication – signup/login, social auth, role-based access. Dynamic data display – lists, repeating groups, conditional visibility. Integrating external APIs using API Connector. (Ref: Rui Barbosa Ch. 4–7)

Textbook Reference:

• Rui Barbosa & Larry Liu – Ch. 4–7
• Webflow University Curriculum (university.webflow.com)

UNIT – III	03 Hours
Workflow Automation & Integrations: Zapier, Make (Integromat), Microsoft Power Automate – triggers, actions, conditions. Multi-step workflows with filters and branching logic. Integrating SaaS tools – Google Sheets, Slack, Gmail, Notion. Webhooks – sending and receiving data. Scheduling workflows and handling errors. (Ref: Micah Laaker, No Code Required Ch. 3–5) Textbook Reference: • Micah Laaker, No Code Required – Ch. 3–5 (O'Reilly, 2022) • Make Academy Documentation (academy.make.com)	
UNIT – IV	**03 Hours**
Mobile App & Backend Development: No-code mobile apps – Glide, Adalo, FlutterFlow overview. Backend-as-a-Service – Airtable, Supabase, Firebase. Push notifications and in-app messaging. Internal tools – Retool and Appsmith for operations dashboards. Form builders – Typeform and JotForm. Connecting LCNC frontend to custom backends via APIs. (Ref: Micah Laaker Ch. 6–9) Textbook Reference: • Micah Laaker, No Code Required – Ch. 6–9 • Firebase Documentation (firebase.google.com/docs)	
UNIT – V	**03 Hours**
AI-Powered LCNC & Deployment: AI-integrated LCNC platforms – Builder.ai, Durable, 10Web. Generating fully functional apps from natural language prompts. OpenAI API integration within LCNC workflows. Testing LCNC Applications – functional and usability testing. Deployment and hosting – custom domains, performance, CDN. Security in LCNC – data privacy, access control, vendor lock-in, and responsible AI deployment. (Ref: OpenAI API Quickstart Docs) Textbook Reference: OpenAI API Quickstart Documentation (platform.openai.com/docs) Shamyla Anwer, The No-Code Playbook (Self-Published, 2023)	

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

NVIDIA Deep Learning Institute (DLI) — FREE Self-Paced Course Mapping
Source: NVIDIA DLI Official Catalog (Aug 2024) + Live URL verification (Apr 2026) | Access: learn.nvidia.com | Only FREE self-paced courses included

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Unit 5	**Generative AI Explained**	Generative AI	1 Hr	—	courses.nvidia.com/course-v1:DLI+S-FX-07+V1	Provides conceptual foundation for AI-powered LCNC tools (Durable, Builder.ai, 10Web) in Unit 5 — students must understand how GenAI works before using AI app generators and integrating OpenAI

						API in LCNC.
Units 4–5	**Prompt Engineering With Llama 2**	Generative AI / LLM	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-FX-12+V1	Prompt engineering strategies directly applicable to generating apps, workflows, and content via AI-powered LCNC platforms and OpenAI API integrations in Units 4–5.
Unit 5	**Augment Your LLM Using RAG**	Generative AI / LLM	2 Hrs	—	courses.nvidia.com/course-v1:NVIDIA+S-FX-16+v1	Introduces RAG pipeline concepts — applicable when students integrate OpenAI API into Bubble.io or Webflow LCNC platforms to build knowledge-base-driven chatbots in Unit 5.
Unit 5	**Building a Brain in 10 Minutes**	Deep Learning	10 Min	—	courses.nvidia.com/course-v1:DLI+T-FX-01+V1	10-min neural network introduction — gives non-CS students (LCNC's primary audience) essential conceptual grounding to understand AI features inside LCNC platforms and evaluate their outputs.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **Prompt-to-App Experiments:** Generate apps from natural language descriptions using AI LCNC tools; critically evaluate output usability, completeness, and accuracy.
- **Reflective Comparison:** Compare AI-assisted vs. manually designed LCNC Applications for quality, customization, and time-to-delivery.
- **AI Automation Critique:** Build AI-suggested workflows and identify edge cases where human judgment, error handling, and override capabilities are essential.
- **Citizen Developer Mindset:** Non-CS students build functional enterprise apps, fostering interdisciplinary collaboration between domain experts and developers.
- **Ethical LCNC Deployment:** Critically examine data ownership, vendor lock-in risks, GDPR compliance, and responsible AI deployment in LCNC platforms.

Applications:

- Business Process Automation for Enterprise HR and Finance
- Startup MVP Development without Engineering Teams
- Internal Tool Building for Operations and Sales TeamsEvent Management and Attendee Registration Portals
- E-Commerce Storefronts and Product Showcases

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Build a task management app with user authentication and role-based access using Bubble.io.
- Design a multi-table relational database and implement CRUD workflows in Bubble.
- Create a multi-step Zapier/Make automation: Google Forms → Google Sheets → Slack notification.
- Build an e-commerce product listing page with search, filter, and cart in Webflow.
- Integrate a third-party REST API (weather/currency exchange) into Bubble using API Connector.
- Develop a mobile app using Glide connected to a Google Sheets backend.
- Create an internal admin dashboard using Retool connected to a live PostgreSQL database.
- Build an AI-powered FAQ chatbot using OpenAI API integration within a LCNC platform.
- Automate a complete business workflow (invoice generation + email dispatch) using Power Automate.
- Deploy a fully functional LCNC web app to a live custom domain; conduct user acceptance testing.

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – LCNC Event Registration Portal

Problem: Complete event management and attendee registration system with payment integration.

Concepts used: Bubble.io, Database design, Workflow automation, Stripe integration

Industry relevance: Event management companies and educational institutions.

Project 2 – Business Workflow Automation Suite

Problem: Automate employee onboarding using Make.com, Airtable, and Slack.

Concepts used: Zapier/Make, Webhooks, API integration, Conditional branching logic**Industry relevance:** HR departments and startup operations teams.

Project 3 – AI-Powered Portfolio Builder

Problem: No-code app that generates portfolio websites from user input using AI content suggestions.

Concepts used: Webflow, OpenAI API, CMS, Custom domain deployment

Industry relevance: Freelancers, students, and creative professionals.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the LCNC paradigm, its market ecosystem, platform types, and applicability across business scenarios.	L2
2	Build functional web Applications using LCNC platforms with database design, authentication, and dynamic UI.	L3
3	Design and implement multi-step workflow automations integrating SaaS tools and external APIs without coding.	L3
4	Develop mobile Applications and internal tools using appropriate LCNC backend and frontend platforms.	L3
5	Evaluate AI-enhanced LCNC tools, deploy production-ready Applications, and analyze security and scalability.	L5

Table: Mapping Levels of COs to POs / PSOs												
	Program Outcomes (POs)										Program Specific Outcomes (PSOs)	
COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PSO 1	PSO 2
CO 1	2	2	2	2	3						2	2	2
CO 2	3	2	3	3	3						2	3	2
CO 3	3	3	2	3	3						2	3	2
CO 4	2	3	3	3	2						2	3	2
CO 5	2	3	2	3	3						2	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools:

- Bubble.io – Visual Web App Builder
- Webflow – No-Code Website Builder
- Make (Integromat) / Zapier – Workflow Automation
- Glide / Adalo – Mobile App Builders
- Airtable / Supabase – Database & Backend
- Microsoft Power Platform – Enterprise LCNC

TEXT BOOKS:

7. Rui Barbosa & Larry Liu, Beginning No-Code Development, Apress, 2022.
8. Micah Laaker, No Code Required, O'Reilly, 2022.
9. Salesforce Trailhead Documentation on Low-Code Development.

REFERENCE BOOKS:

4. Shamyla Anwer, The No-Code Playbook, Self-Published, 2023.
5. OutSystems, Low-Code Development Guide, OutSystems, 2023.

E-RESOURCES:

- https://bubble.io/academy
- https://university.webflow.com
- https://academy.make.comhttps://learn.nvidia.com (NVIDIA DLI Free Courses)

Activity Based Learning (Suggested Activities in Class)

- Build-a-thon – 48-hour no-code app challenge in teams of 3–4 students.
- Platform showcase – each student presents an independently built LCNC app.
- Industry case study: analyze how startups (Zapier, Notion) bootstrapped using LCNC approaches.
- Guest lecture from a no-code developer or citizen developer working in an enterprise context.

****

DIFFERENTIAL EQUATIONS & OPTIMIZATION

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - III

Course Code	:		**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	30 Hrs L
L–T–P–J	:	3–0–0–0			

Course Vision:

To develop strong mathematical and computational foundations in vector calculus, numerical methods and optimization techniques for solving complex engineering problems and designing intelligent AI-driven optimization systems.

Course Objectives:

This course will enable students to:

1. Understand and apply vector calculus concepts for modelling engineering and scientific systems.
2. Apply numerical methods for solving ordinary and partial differential equations arising in dynamic systems.
3. Analyze and solve classical optimization problems encountered in engineering applications.
4. Develop problem-solving skills for AI and computational systems.
5. Apply nonlinear optimization algorithms to solve complex real-world optimization problems.

Broad Course Description:

The course Differential Equations & Optimization provides a comprehensive foundation in vector calculus, numerical methods, and optimization techniques used to solve complex engineering and computational problems. It covers concepts such as gradient, divergence, curl, numerical solutions of ordinary and partial differential equations, linear programming, and nonlinear optimization methods including gradient descent and penalty methods. The course emphasizes practical problem-solving using Python and MATLAB to model dynamic systems, optimize engineering designs, and support AI-driven decision-making systems. Through theoretical understanding and computational implementation, students develop analytical and optimization skills applicable to robotics, machine learning, control systems, resource allocation, and intelligent engineering applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various

Course Outcomes:

1. Lecture Method**:** Concepts, algorithms, and models are explained through structured lectures supported by Python/MATLAB demonstrations.
2. Interactive Teaching**:** Active learning is promoted through discussions, brainstorming, and group problem-solving activities.
3. Demonstration using Python/MATLAB**:** AI techniques like classification and optimization are demonstrated to visualize results and understand implementation.
4. Multimedia Learning: Videos and simulations are used to explain complex concepts and real-world Applications.
5. Collaborative Learning: Students work in groups to implement projects and analyze datasets using computational tools.
6. Higher Order Thinking Questions: Analytical questions are used to develop model design, evaluation, and decision-making skills.

7. Problem-Based Learning: Real-world problems are given to apply techniques using Python/MATLAB for practical solutions.	
MODULE – I	**06 Hours**
Vector Calculus **Topics:** Introduction to vector calculus, scalar and vector valued functions, gradient, tangent plane, directional derivative, divergence and curl, scalar and vector potentials. Statement of vector identities and simple problems. Line, surface and volume integrals. Statement of Green's theorem, Stokes' theorem and Gauss divergence theorem. **Applications:** Fluid flow analysis, Electromagnetic field modelling, Robotics motion planning, Computer graphics and physical simulations	
MODULE – II	**06 Hours**
Numerical Methods for ODE and PDE **Topics:** Existence and uniqueness of solutions for ordinary differential equations. Solving initial value problems using Taylor series method, Euler's method and modified Euler method, Runge-Kutta methods, multistep methods. Higher-order equations and systems. Stability analysis. Solving boundary value problems using shooting methods, difference methods and variational methods. Introduction to numerical solutions for partial differential equations. **Applications:** Weather prediction models, Mechanical system simulations, Control system modelling, Heat transfer and diffusion problems.	
MODULE – III	**06 Hours**
Introduction to Optimization (Classical Optimization) **Topics:** Introduction to classical optimization, engineering Applications of optimization, classification of optimization problems, single variable optimization, multivariable optimization without constraints, multivariable optimization with equality and inequality constraints, Kuhn-Tucker conditions. **Applications:** Resource allocation problems, Engineering design optimization, Production planning, Logistics and supply chain optimization	
MODULE – IV	**06 Hours**
Linear Programming **Topics:** Linear Programming: Linear programming, standard form of linear programming, geometry of linear programming problems, solution of a system of linear simultaneous equations, pivotal production of general systems of equations, graphical method, simplex algorithms, revised simpler methods, duality in linear programming. **Applications:** Resource allocation and production planning, Transportation and logistics optimization, Supply chain management, Financial portfolio optimization, Scheduling and assignment problems	
MODULE – V	**06 Hours**
Nonlinear Optimization **Topics:** Unconstrained nonlinear optimization, univariate method, gradient of a function, gradient descent method, Fletcher-Reeves method. Constrained nonlinear optimization, characteristics of a constrained optimization problem, cutting plane method, interior and exterior penalty function methods. **Applications:** Neural network training, Portfolio optimization, Control system optimization, Engineering system design	

Authentic Intelligence:

1. Understanding limitations of mathematical approximations in AI systems
2. Evaluating numerical errors in computational models
3. Ethical considerations in automated optimization systems

4. Ethical decision-making in optimization problems
5. Understanding biases introduced by optimization algorithms

AI Integration:

1. Gradient calculations in machine learning optimization
2. Simulation of dynamic systems in robotics
3. Decision optimization in intelligent systems
4. Linear programming used in decision-making systems
5. Gradient descent optimization algorithms for training neural networks and deep learning

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply vector calculus concepts such as gradient, divergence, curl, and vector integrals to solve engineering and computational problems.	L3
2	Implement numerical methods for solving ordinary and partial differential equations arising in engineering and scientific models.	L3
3	Formulate and solve classical optimization problems including single and multivariable optimization with constraints.	L4
4	Develop and solve linear programming models for real-world problems	L3
5	Apply nonlinear optimization techniques such as gradient descent, Fletcher–Reeves method, and penalty methods to engineering Applications.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

	Program Outcomes (POs)											PSOs	
COS	1	2	3	4	5	6	7	8	9	10	11	1	2
CO-1	3	2			2					1		2	–
CO-2	3	3	2	2	3					1		3	1
CO-3	3	3	3	2	3				1	2		3	2
CO-4	3	3	3	2	2					1		3	2
CO-5	3	3	2	2	3	1		1		1	2	2	2

3: Substantial (High)   2: Moderate (Medium)   1: Poor (Low)

TEXT BOOKS:

1. Hamdy A. Taha, Operations Research: An Introduction, 2017, 10^{th} Edition, Pearson Education, Inc.
2. Rao, S.S., Engineering optimization: theory and practice, 2019, 5^{th} Edition,

John Wiley & Sons, Inc.

REFERENCE BOOKS:

1. Arora, R.K., Optimization: algorithms and Applications, 2015, 1st Edition, Chapman and Hall/CRC.
2. Deb, K., Optimization for engineering design: Algorithms and examples, 2012-2nd Edition, PHI Leaming Pvt. Ltd.
3. Nocedal, Jorge, and Stephen J. Wright. *Numerical optimization*. New York, NY: Springer New York, 2006.

E-RESOURCES:

RECOMMENDED TOOLS

- Python
- MATLAB
- NumPy
- Pandas
- Matplotlib
- Scikit-learn

LAB EXERCISES

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Vector Calculus (03 Hours)					
1	Gradient Computation	Introduction to computational tools and compute gradient of scalar functions	MATLAB/PYTHON	1 h	**CO1**
2	Tangent Plane & Directional Derivative	Compute divergence and curl of vector fields, Visualize tangent planes and directional derivatives	MATLAB/PYTHON	1 h	**CO1**
3	Line Integrals	Implement numerical computation of line integrals	MATLAB/PYTHON	1 h	**CO1**
MODULE II: Exercises 4–6: Numerical Methods for ODE & PDE (03 Hours)					
4	Taylor Series and Euler Method	Solve ODE using Taylor series method, Euler method and modified Euler method	MATLAB/PYTHON	1 h	**CO2**
5	Runge-Kutta and Multistep Method	Implement fourth-order Runge-Kutta method and multistep method	MATLAB/PYTHON	1 h	**CO2**
6	Shooting and Finite Difference Method	Solve boundary value problems using shooting and finite difference method	MATLAB/PYTHON	1 h	**CO2**
MODULE III: Exercises 7–9: Introduction to Optimization (03 Hours)					
S.No	**Exercise Title**	**Aim / Objective**	**Tools**	**Hrs**	**CO**
7	Single Variable Optimization	Implement optimization for single variable functions	MATLAB/PYTHON	1 h	**CO3**
8	Multivariable optimization	Perform multivariable optimization and their visualization	MATLAB/PYTHON	1 h	**CO3**
9	Kuhn-Tucker Conditions	Implement KKT conditions for constrained optimization	MATLAB/PYTHON	1 h	**CO3**

MODULE IV: Exercises 10–12: Linear Programming (03 Hours)					
10	Formulation of LPP	Formulate and solve linear programming problems (LPP) and implement graphical method	MATLAB/PYTHON	1 h	**CO4**
11	Simplex Method, Revised Simplex Method	Implement simplex algorithm for LPP, Solve LPP using revised simplex method	MATLAB/PYTHON	1 h	**CO4**
12	Duality in LPP	Analyze duality concepts in linear programming	MATLAB/PYTHON	1 h	**CO4**
MODULE V: Exercises 13–15: Nonlinear Optimization (03 Hours)					
13	Fletcher-Reeves Method	Implement Fletcher-Reeves optimization method	MATLAB/PYTHON	1 h	**CO5**
14	Cutting Plane Method	Apply cutting plane method for constrained optimization	MATLAB/PYTHON	1 h	**CO5**
15	Interior and Exterior Penalty Method	Implement interior and Exterior penalty function method	MATLAB/PYTHON	1 h	**CO5**

****

Operating Systems for GPUs

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours L
L–T–P-J	:	3–0–0-0			

Course Vision: This course aims to develop a strong understanding of operating system principles and their critical role in modern AI and high-performance computing systems. It aims to enable students to connect fundamental OS concepts such as processes, memory management, scheduling, and synchronization with real-world AI workloads running on GPUs and distributed systems. The course also aims to equip learners with practical skills to design, analyze, and optimize scalable and efficient AI infrastructure by integrating operating system concepts with CUDA programming, concurrency models, and system-level observability tools.

Prerequisites: Linux Fundamentals, Programming in C Language

Course Learning Objectives:

This Course will enable students to:

1. Explain and differentiate core operating system concepts such as processes, threads, scheduling algorithms, memory management, file systems, and synchronization.
2. Apply scheduling principles to GPU job scheduling, mapping CPU algorithms to CUDA kernel launches and evaluating NVIDIA MPS for concurrent AI workloads.
3. Analyze memory management techniques (paging, segmentation, TLB) in the context of CUDA's memory hierarchy and unified memory for efficient AI training.
4. Evaluate file system structures (ext4, NTFS, journaling, B+ trees) for storing and checkpointing large-scale LLM weights on high-performance storage systems.
5. Implement synchronization mechanisms (mutexes, semaphores, monitors) to design concurrent inference servers and avoid deadlocks in GPU resource contention.
6. Integrate OS-level concurrency and containerization (Python threading, multiprocessing, asyncio, namespaces, cgroups) to build scalable, isolated AI workloads.
7. Utilize observability tools (strace, perf, NVIDIA NSight Systems) to profile, monitor, and optimize end-to-end inference pipelines in distributed AI systems.

Broad Course Description

This course bridges core operating system principles with modern AI infrastructure, showing how processes, threads, scheduling, memory management, file systems, and synchronization directly impact large-scale AI workloads. Students will learn classical concepts such as CPU scheduling algorithms, paging, segmentation, journaling, and concurrency, while applying them to GPU job scheduling, CUDA memory hierarchy, NVMe-based checkpointing, and thread pools for inference servers. The curriculum emphasizes practical tie-ins like simulating GPU schedulers, mapping CPU scheduling to CUDA kernel launches, and analyzing NVIDIA MPS for concurrent streams. It also explores container internals, Python concurrency models, and observability tools for profiling end-to-end inference pipelines. By integrating textbook theory with case studies on LLM batch inference, distributed training, and GPU resource management, the course equips learners to design efficient, scalable, and resilient AI systems grounded in OS fundamentals.

UNIT – I	**09 Hours**

Processes, Threads, Scheduling Concepts: Processes vs. threads, scheduling algorithms (FCFS, SJF, Round Robin, Priority).

Textbook-1 Coverage: 3.1–3.3, 4.1, 4.2, 4.4, 5.1–5.3.

Map CPU scheduling to GPU kernel launches.

Compare Round Robin vs. NVIDIA MPS scheduling for concurrent CUDA streams. Case study: Batch inference scheduling for LLM workloads.

Textbook-2 Coverage: 3.1–3.2, 3.4, 3.7	
UNIT – II	**09 Hours**
Memory Management: Concepts: Paging, segmentation, virtual memory, TLB. Textbook-1 Coverage: 8.1–8.6. GPU global vs. shared vs. texture memory. CUDA Unified Memory bridging CPU–GPU address spaces. TLB relevance in GPU page migration. Textbook-3 Coverage: 5.4, 5.5	
UNIT – III	**09 Hours**
File Systems Concepts: ext4, NTFS, journaling, B+ tree ndexing. Textbook-1 Coverage: 10.1–10.6. Checkpointing large LLM weights (multi-TB). Journaling for crash recovery in distributed training.	
UNIT – IV	**09 Hours**

Synchronization Concepts: Mutex, semaphore, monitors, deadlock. Textbook-1 Coverage: 6.1–6.7. Avoiding deadlocks in GPU resource contention. Monitor-based synchronization for async inference pipelines. Textbook-4 Coverage: 13.6

UNIT – V	**09 Hours**
OS for AI Infrastructure Concurrency in Python: threading, multiprocessing, asyncio, concurrent.futures → async LLM inference endpoint. Container Internals: namespaces, cgroups, copy-on-write → Docker isolation for AI workloads. GPU Scheduling: NVIDIA MPS enabling concurrent CUDA contexts on DGX B200. Observability: strace, perf, NVIDIA NSight Systems → profiling end-to-end inference workloads. Source: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-OV-16+V1	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

The integration of AI systems is framed through classical operating system concepts. Traditional OS mechanisms—such as processes, threads, scheduling, memory management, file systems, and synchronization—are directly mapped to modern AI infrastructure. For example:

- Scheduling: CPU scheduling algorithms are compared to GPU kernel launches and NVIDIA MPS handling of concurrent CUDA streams.
- Memory management: Paging and TLBs are linked to CUDA's unified memory and GPU page migration.
- File systems: Structures like journaling and B+ trees are contextualized through large-scale LLM checkpointing on NVMe RAID.
- Synchronization: Classical primitives are applied to thread pools serving concurrent inference requests.Concurrency & observability: OS concurrency models, container internals, GPU scheduling, and profiling tools are integrated to optimize AI inference workloads.

Applications:

The course bridges academic theory with practical AI infrastructure engineering.

Students learn to apply classical networking and distributed systems concepts to modern AI workloads. Key Applications include:

- **Network optimization**: Using latency analysis and network models to improve LLM streaming APIs.
- **Cloud architecture**: Designing multi-region topologies with efficient routing and secure communication via HTTP/2, WebSockets, and gRPC.
- **Global distribution**: Leveraging DNS, CDN, and load balancing for worldwide AI services.
- **Consistency & databases**: Applying CAP theorem and consensus algorithms to vector databases and parameter servers.
- **GPU communication**: Benchmarking frameworks like NVIDIA NCCL for high-throughput distributed training.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	**Apply classical network and OS concepts** (models, protocols, scheduling, memory, synchronization) to design and optimize modern AI infrastructure	L3
CO2	**Analyze latency and scalability challenges** in AI APIs, implementing solutions with HTTP/2 multiplexing, CDN distribution, and load balancing	L3
CO3	**Design and evaluate distributed systems** for AI workloads, applying CAP theorem, consensus algorithms, and GPU scheduling strategies.	L3
CO4	**Implement and benchmark communication models** (REST vs gRPC, SSE, WebSockets) and GPU frameworks (NCCL, NVLink) for high-performance inference and training.	L3
CO5	**Integrate observability and reliability practices** (profiling, journaling, deadlock avoidance) to ensure efficient, resilient, and concurrent AI service deployment.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	3							2	3	
CO2	3	3	3	3							2	3	
CO3	3	3	2	3							2	3	
CO4	3	3	3	3							3	3	
CO5	3	3	3	3							3	3	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2010
2. NVIDIA Ebook 3rd-Edition-Chapter03-scalable-parallel-execution
3. NVIDIA Ebook 3rd-Edition-Chapter05-performance-considerations
4. NVIDIA Ebook 3rd-Edition-Chapter13-cuda-dynamic-parallelism

E-RESOURCES:

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-AC-01+V1

Activity Based Learning (Suggested Activities in Class)

1. Build a multithreaded REST API server (Python + FastAPI) that serves an LLM and an image classifier concurrently. Benchmark: throughput (req/s), latency (P50, P90, P99),GPU utilisation (nvidia-smi). Apply OS profiling tools. Frame as "production readiness report" for a startup deploying its first AI service.

****

Design and Analysis of Algorithms

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	60 Hrs (45Hrs L+15 Hrs P)
L–T–P-J	:	3–0–2-0			

Course Vision: This course aims to develop strong analytical and problem-solving skills in students by introducing fundamental algorithm design techniques and efficiency analysis methods. It aims to enable learners to design, analyze, and optimize algorithms using various paradigms such as divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound. The course also aims to build a strong mathematical and logical foundation for solving computational problems efficiently in real-world applications.

Prerequisites: Data Structures Using C and Programming Fundamentals.

Course Learning Objectives:

This Course will enable students to:

1. Analyze the non-recursive and recursive algorithms and to represent efficiency of these algorithms in terms of the standard Asymptotic notations.
2. Devise the Brute Force and Divide and Conquer techniques to design the algorithms and apply these methods in designing algorithms to solve a given problem.
3. Explain the Decrease and Conquer, Transform and Conquer algorithm design techniques, and Time versus Space Trade-offs.
4. Get the idea of Greedy method and dynamic programming methods and apply these methods in designing algorithms to solve a given problem.
5. Describe and illustrate the idea of Backtracking and Branch and Bound algorithm design techniques to solve a given problem.

Broad Course Description

This course offers a systematic approach to algorithm design and efficiency analysis. It links data structures with algorithmic paradigms, establishing a strong foundation for domains such as AI optimization, machine learning model tuning, cybersecurity algorithms, and data analytics. Students develop mathematical reasoning and logical rigor to analyze both exact and approximate solutions.

UNIT – I	**09 Hours**

INTRODUCTION:

What is an Algorithm? Fundamentals of Algorithmic Problem Solving. (Text Book-1: Chapter 1: 1.1 to 1.2)

FUNDAMENTALS OF THE ALGORITHMS' EFFICIENCY:

Analysis Framework, Asymptotic Notations and Standard notations and common functions
(Text Book-2: Chapter 3: 3.1, 3.2),
Mathematical Analysis of Non-recursive and Recursive Algorithms,
(Text Book-1: Chapter 2: 2.1, 2.3, 2.4,)

UNIT – II	**09 Hours**

BRUTE FORCE:

Background, Selection Sort, Brute-Force String Matching. (Text Book-1: Chapter 3: 3.1, 3.2)

DIVIDE AND CONQUER:

General method, Recurrences: The recursion-tree method, The master method. (Text Book-2: Chapter 4: 4.4, 4.5),
Merge sort, Quick sort, Binary Search, Multiplication of large integers,

Case study:

(Text Book-1: Chapter 4: 4.1 to 4.3, 4.5)

UNIT – III	09 Hours
DECREASE & CONQUER: General method, Insertion Sort, Graph algorithms: Depth First Search, Breadth First Search, Topological Sorting **TRANSFORM AND CONQUER:** Case study: Heaps and Heap sort. **TIME AND SPACE TRADEOFFS:** Input Enhancement in String Matching: Horspool's algorithm, Hashing: Open and Closed hashing. (Text Book-1: Chapter 5: 5.1 to 5.3, Chapter 6: 6.3 to 6.4, Chapter 7: 7.2 to 7.3)	

UNIT – IV	09 Hours
GREEDY TECHNIQUE: General method of Greedy technique, Single-Source Shortest Paths: General method, The Bellman-Ford algorithm, Single-Source Shortest Paths in DAGs, Dijkstra's Algorithm (Text Book-2: Chapter 24: 24.1 to 24.3). Minimum Spanning Trees: Prim's Algorithm, Optimal Tree Problem: Huffman Trees. Case study: Kruskal's Algorithm (Text Book-1: Chapter 9: 9.1, 9.2, 9.4). **DYNAMIC PROGRAMMING:** General method, The Floyd- Warshall Algorithm, Johnson's algorithm for sparse graphs (Text Book-2: Chapter 25: 25.1 to 25.3), The Knapsack problem (Text Book-1: Chapter 8: 8.4).	

UNIT – V	09 Hours
LIMITATIONS OF ALGORITHMIC POWER: P, NP and NP-complete problems (Text Book-1: Chapter 11: 11.3) **BACKTRACKING:** General method, N-Queens problem, Subset-sum problem. (Text Book-1: Chapter 12: 12.1) **BRANCH AND BOUND:** General method, Travelling Salesman problem, Approximation algorithms for TSP. **Case study:** Knapsack Problem. (Text Book-1: Chapter 12: 12.2, 12.3)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Algorithmic thinking for AI model optimization
- Dynamic programming and greedy strategies in reinforcement learning
- Graph algorithms in neural network and social network connectivity
- Heuristic search for game and AI decision systems
- Responsible algorithmic design (fairness, bias minimization)

Applications

1. Search engines – optimizing query processing and keyword indexing.
2. Routing and navigation systems – computing shortest or fastest paths.
3. Scheduling and resource allocation – optimizing CPU, job, or network tasks.
4. Data compression – efficient encoding using greedy and dynamic algorithms.
5. Artificial intelligence and machine learning – applying search and optimization algorithms.
6. Cybersecurity – cryptographic key generation and attack-path analysis.

Laboratory Components (1 Credit)

The lab emphasizes **hands-on implementation, visualization, and AI-assisted debugging.**

The following are experiments to be carried out using either C programming language or the object-oriented programming language:

Experiment 1

1a) Write a C program function to perform a linear search on a list of integers. The function should take two parameters - the array of integers and the target value to be searched. If the target value is present in the list, the function should return all the index of the occurrence of the target value. If the target value is not found, print Not found message.

1b) Write a C program that includes two functions. The first function, named validateSortedArray, should take an array of integers as input and validate if the array is sorted in ascending order. The second function, named binarySearch, should perform binary search on a sorted list of integers. The binarySearch function should take two parameters – the validated sorted list of integers and the target value to be searched. If the target value is present in the list, the function should return the index of the target value. If the target value is not found, the function should return -1. Implement both functions and provide an example of their usage.

Experiment 2

Write a C program that implements the Merge Sort algorithm to sort an array of integers in ascending order. Create a function mergeSort that takes an array and its size as parameters and sorts the array using the Merge Sort technique. Additionally, create a function merge to merge two sorted halves of the array during the sorting process. Finally, print the sorted array.

Experiment 3

Write a C program that utilizes the array data structure to determine whether a given undirected graph is connected or not using the Depth-First Search (DFS) method. The program should take input for the number of vertices, the number of edges, and the edges themselves. It should then use DFS to traverse the graph and output whether the graph is connected or not.

Experiment 4

You are given an undirected graph represented by an adjacency matrix. Write a C Progeam to perform Breadth-First Search (BFS) traversal starting from a given vertex. The function should return the order in which the vertices are visited during the traversal.

Experiment 5

Implement Warshall's Algorithm to determine the transitive closure of a given directed graph represented as an adjacency matrix. The transitive closure of a graph determines all the vertices reachable from every other vertex in the graph, considering all possible paths. Warshall's Algorithm efficiently computes this closure using dynamic programming. The program should prompt the user to input the directed graph as an adjacency matrix and then apply Warshall's Algorithm to compute the transitive closure. Finally, print the resulting transitive closure matrix.

Experiment 6 Develop an efficient implementation of Floyd's algorithm to determine the shortest paths between all pairs of vertices in weighted graph. Floyd's Algorithm efficiently computes this using dynamic programming. The input should be a weighted adjacency matrix representing the graph, and the output should be the shortest distance matrix showing the shortest path distances between every pair of vertices.

Experiment 7

You are a thief planning to rob a store. The store contains various items, each with its own weight and value. However, your knapsack can only carry a limited weight capacity. Given the weights and values of each item, determine the maximum total value you can obtain without exceeding the weight capacity of your knapsack.

Experiment 8

You're tasked with implementing Kruskal's algorithm in the C programming language to find the Minimum Cost Spanning Tree (MCST) for a given weighted graph. The algorithm should efficiently determine the subset of edges that connect all vertices of the graph with the minimum total weight. Design two functions called find and union. The find function is used to determine the representatives of the sets containing the source and destination vertices of the edge. The Union function is used to merge two disjoint sets into a single set by connecting their representatives.

Experiment 9

You're tasked with implementing Dijkstra's algorithm in the C programming language to find the Single Source Shortest Path for a given weighted graph. The algorithm should efficiently determine shortest path from one specific vertex to all other vertices in a graph, optimizing the total distance traveled.

Experiment 10

Open-ended programs

i) Implement N-Queens problem using Backtracking technique
ii) Write a C program to implement Prim's Algorithm to find the Minimum Spanning Tree (MST) of a connected, undirected, weighted graph. The program should accept the number of vertices and the weighted adjacency matrix as input, and output the edges selected for the Minimum Spanning Tree along with the total minimum cost.
iii) Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Demonstrate this algorithm using the Divide-and-Conquer method.
iv) Case study

Mini Project Component

Industry-Driven Projects (Google / Amazon Style)

Some sample projects described below simulate **real software-engineering problems** and align with algorithm design, analysis, and optimization practices used in the industry.

Project 1 – Search Engine Indexing

- **Problem:** Design a data structure and algorithm that efficiently stores, searches, and retrieves web search queries or documents. Create indexing and ranking based on keyword frequency and relevance.

- **Concepts used:**Divide and Conquer, Sorting (Quick Sort / Merge Sort), Hash Tables, T rees (BST/Trie), Search Optimization Algorithms.
- **Industry relevance:** Core component of search engines such as Google and DuckDuckGo for high-performance data retrieval.

Project 2 – Social Network Graph Analyzer

- **Problem:** Build a system that models and analyzes a social network to identify key influencers, compute shortest connection paths, and find communities or clusters.
- **Concepts used:** Graph Algorithms (BFS, DFS, Dijkstra, Minimum Spanning Tree), D ynamic Programming for path optimization, Complexity Analysis.
- **Industry relevance:** Used in social and professional platforms like Meta and LinkedIn for friend recommendation and network insight generation.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Analyze the given recursive and non-recursive algorithms for time using step count, substitution method, recurrence tree method and mathematical analysis method. Represent the complexity of the algorithm using asymptotic notation.	L3
CO2	Solve sorting, searching, matrix multiplication problems based on divide and conquer design technique and implement in C programming language.	L3
CO3	Apply algorithms for graph-based problems (DFS, BFS and Topological sorting) using decrease and conquer design techniques. Distinguish the trade-offs between space and time complexity.	L3
CO4	Apply algorithms for finding the shortest path and minimum spanning tree for a given graph using greedy and dynamic programming techniques and implement the 0/1 knapsack problem in C programming language.	L3
CO5	Apply an efficient algorithm to solve N-Queens problem, Subset-sum problem, Knapsack and Traveling salesman problem using branch and bound and backtracking design technique. Describe the limitation of algorithmic power in terms of P, NP, NP hard and NP complete categorizations.	L3
CO6	Implement the graph-based algorithms including DFS, BFS, Warshall's Algorithm, Floyd's Algorithm, Kruskal's Algorithm, and Dijkstra's Algorithm in C programming language. (Lab Experiments)	L3

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	2	3								1		3	
CO2	2				3					1		3	
CO3	2	2	3	3						1		2	
CO4	2	2	3		3					1		3	

CO5	2	2	3							1		2	
CO6	2	2	3		3					1		3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Anany Levitin, "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2011.
2. Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI, 2014.

REFERENCE BOOKS:

1. Horowitz E., Sahni S., Rajasekaran S, "Computer Algorithms", Galgotia Publications, 2001.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T.Tsai, "Introduction to the Design and Analysis of Algorithms: A Strategic Approach", Tata McGraw Hill, 2005.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Fundamentals of Accelerated Computing with CUDA	III–V	Optimization-based algorithm acceleration
AI Workflow: Algorithmic Efficiency and Model Optimization	IV–V	Real-world impact of algorithm complexity
Intro to Graph Neural Networks	III	Extension of graph algorithms

E-RESOURCES:

1. https://nptel.ac.in/courses/106/101/106101060/
2. http://cse01-iiith.vlabs.ac.in/
3. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAl gor ithms
4. www.nvidia.com
5. https://www.coursera.org/specializations/algorithms

Activity-Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification, Cabbage puzzle, Konigsberg bridge puzzle etc.,
2. Demonstration of solutions to a problem through programming.

****

MACHINE LEARNING
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - IV

Course Code	:		**Credits**	:	04
Hours / Week	:	05	**Total Hours**	:	30 Hrs L + 15 Hrs P
L–T–P–J	:	3–0–2–0			

Course Vision:
To develop the ability to design, analyze, and implement machine learning solutions for engineering problems by integrating data-driven modeling, computational tools, and responsible AI practices.

Course Objectives:
This course will enable students:
1. To summarize the basic concepts, Feature Engineering and different types of Machine Learning.
2. To make Use of the Supervised Machine Learning techniques for solving appropriate real-world applications.
3. To make Use of the Unsupervised Machine Learning techniques, Dimensionality Reduction techniques for solving appropriate real-world applications.
4. To utilize the Probabilistic Graphical Models and Ensemble Learning Methods for handling complex real time problems.
5. To evaluate the performance and reliability of machine learning models.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:
1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I — **09 Hours**

Foundations of Machine Learning

Topics: Well-posed learning problems, Designing a Learning system, Introduction to Machine learning and types Of Learning: Supervised, Unsupervised, And Reinforcement Learning, Data Pre-processing: Missing Value Imputation, Handling balanced and Imbalanced Data, and Feature Engineering.

Applications: Classification and regression problems in real-world scenarios, predictive analytics in healthcare and finance, feature engineering for improving model performance.

MODULE – II — **09 Hours**

Supervised Machine Learning Algorithms

Topics: Introduction to Supervised Learning: Binary and Multi-Class Classification, Regression - Linear Regression and Nonlinear Regression, Logistic Regression, Naive Bayes Classification, Decision Trees (ID3, CART) and Random Forest, K-Nearest Neighborhood, Support Vector Machine (Linear and Non-linear Kernels).

Applications: house price prediction systems, medical diagnosis systems, fault detection in machines, financial risk prediction.

MODULE – III — **09 Hours**

Unsupervised Machine Learning Algorithms **Topics:** Introduction to Unsupervised Learning, Clustering (Hard and Soft Clustering), Hierarchal Clustering: K-Means, Fuzzy C-Means (FCM) Algorithm, Gaussian Mixture Models (GMM), Expectation Maximization Algorithm, Dimensionality Reduction - Linear Discriminant Analysis and Principal Component Analysis. **Applications:** customer segmentation in business analytics, anomaly detection in network traffic, pattern discovery in industrial data.	
MODULE – IV	**09 Hours**
Probabilistic Graphical Models and Ensemble Learning **Topics:** Probabilistic Graphical Models - Bayesian Networks, Markov Random Fields, and Ensemble Learning - Bagging Algorithms, Understanding Out-Of-Bag Score, Boosting: AdaBoost, Gradient Boosting Machines (GBM), XGBoost, CatBoost. **Applications:** uncertainty modeling using Bayesian Networks, image analysis with Markov Random Fields, prediction tasks using ensemble methods, fraud detection and recommendation systems with boosting algorithms.	
MODULE – V	**09 Hours**
Machine Learning Model Validation, Optimization, and Regularization **Topics:** Hyper parameter Tuning: Grid Search CV, Random Search CV, Machine Learning Model Validation - Confusion Matrix, Accuracy, Precision, Recall, F1 Score, Cost Function, Cross-Validation, Error Metrics (MSE, RMSE, MAE). Machine Learning Optimization Algorithms: Gradient Descent Techniques, Bayesian Optimization, Overfitting, Underfitting, Optimal Fit, Bias, Variance, Regularization - L1 Regularization, L2 Regularization, Elastic Net Regularization. **Applications:** model tuning for improved prediction accuracy, performance evaluation in classification tasks, optimization of machine learning models in healthcare, finance, and engineering applications.	

Authentic Intelligence:

1. Understanding well-posed learning problems and their real-world constraints, challenges in designing efficient learning systems, limitations of deep learning models, risks associated with poor feature selection and sampling bias, ethical considerations and responsible deployment of AI systems.
2. Interpretability of machine learning models, risks of incorrect predictions in automated decision-making systems
3. Risks of misinterpreting clusters, responsible interpretation of patterns discovered from data.
4. Limitations of graphical and ensemble models, overfitting risks in boosting, interpretability challenges, and importance of responsible model selection.
5. Understanding overfitting and underfitting, bias-variance tradeoff, limitations of optimization methods, and importance of selecting appropriate validation and regularization techniques.

AI Integration:

1. Introduction to Python-based machine learning workflow, data preprocessing and feature engineering techniques, usage of libraries like NumPy, Pandas, and Scikit-learn for building and evaluating models.
2. Implementation of supervised learning algorithms using Scikit-learn, training and evaluating models using real-world datasets
3. Implementing clustering and dimensionality reduction techniques using Scikit-learn, visualization of clusters using Python tools.
4. Implementation of graphical models and ensemble techniques using Scikit-learn, use of bagging and boosting methods, model evaluation with Out-of-Bag score.
5. Use of Grid Search and Random Search for hyperparameter tuning, implementation of validation techniques and performance metrics, application of optimization algorithms and regularization methods

using Scikit-learn.

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Describe the basic concepts and different types of Machine Learning techniques and to design Well Posed Learning System.	L2
2	Apply the learned concepts of the Supervised learning algorithms for regression and classification problems.	L3
3	Apply Unsupervised Machine Learning Algorithms such as Hard and Soft clustering and Dimensionality Reduction techniques for solving appropriate real-world applications.	L3
4	Make Use of the learned concepts of the Probabilistic Graphical Models and Ensemble Learning Methods for handling complex real time problems.	L3
5	Evaluate the performance of Machine Learning algorithms using appropriate metrics such as accuracy, precision, recall, FI Score and Make Use of the different Optimization and Regularization Techniques for improving the model performance.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)										PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	1	-	-	-	-	-	-	-	1	2	
CO2	2	2	1	-	-	-	-	-	-	-	1	2	
CO3	3	2	1	1	1	2	1	-	2	1	1	2	
CO4	3	2	1	1	1	2	1	-	2	1	1	2	
CO5	3	2	1	1	1	2	1	-	2	1	1	2	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Machine Learning – Tom M. Mitchell, McGraw-Hill, 1st Edition, 1997.
2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron, O'Reilly Media, 3rd Edition, 2022.

REFERENCE BOOKS:

1. Pattern Recognition and Machine Learning – Christopher M. Bishop, Springer, 1st Edition, 2006.
2. Introduction to Machine Learning with Python – Andreas C. Müller & Sarah Guido, O'Reilly Media, 1st Edition, 2017.

E-RESOURCES:

1. **Accelerating End-to-End Data Science Workflows**
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-01+V2
2. **Best Practices in Feature Engineering for Tabular Data With GPU Acceleration**
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-06+V1
3. **Accelerating Clustering Algorithms to Achieve the Highest Performance**
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-04+V1
4. **Exploring Adversarial Machine Learning**
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-03+V1
5. **Accelerate Data Science Workflows with Zero Code Changes**
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-03+V1

RECOMMENDED TOOLS

- Programming: Python (NumPy, Pandas, Scikit-learn, PyTorch) – Google Colab/ Jupyter Notebook/Visual Studio
- Visualization: Matplotlib, Seaborn, Plotly, UMAP
- Experiment Tracking: MLflow, Weights & Biases
- MLOps & Deployment: Docker, FastAPI, ONNX
- AI Assistants: GitHub Copilot, ChatGPT, Claude

LAB EXERCISES

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I · Exercises 1–3 : Foundations of Machine Learning					**(06 Hours)**
1	Data Analysis and Visualization	Perform Exploratory Data Analysis and Visualize Features using the Iris Dataset	Google Colab/ Jupyter	2 h	CO1
2	Handling Missing Values, Scaling Features	Preprocess Data by Handling Missing Values, Encoding Categorical Variables, and Scaling Features on Standard Dataset.	Google Colab/ Jupyter	2 h	CO1
3	Split Dataset, cross validation	Split Dataset into Training and Testing Sets and Apply Cross-Validation.	Google Colab/ Jupyter	2 h	CO1
MODULE II · Exercises 4–9 : Supervised Machine Learning Algorithms					**(12 Hours)**
4	Linear Regression	Implement Linear Regression to Predict	Google Colab/	2 h	CO2

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
		House Prices using the Housing Price Dataset.	Jupyter		
5	Logistic Regression	Build a Logistic Regression Model to Perform Classification	Google Colab/ Jupyter	2 h	CO2
6	K-Nearest Neighbors (KNN)	Apply the K-Nearest Neighbors (KNN) Algorithm to Classify Flower Species using the Iris Dataset	Google Colab/ Jupyter	2 h	CO2
7	Decision Tree Model	Train and Visualize a Decision Tree Model to Classify Wine Quality using the Wine Quality Dataset.	Google Colab/ Jupyter	2 h	CO2
8	Random Forest Model	Develop a Random Forest Model to Predict Heart Disease using the Heart Disease Dataset	Google Colab/ Jupyter	2 h	CO2
9	Linear and Non-linear Support Vector Machine	Implementation of Linear and Non-linear Support Vector Machine for Classification	Google Colab/ Jupyter	2 h	CO2
MODULE III ·Exercises 10–11 :Unsupervised Machine Learning Algorithms (04 Hours)					
10	K-Means Clustering	Apply K-Means Clustering to Perform Customer Segmentation using the Mall Customer Segmentation Dataset.	Google Colab/ Jupyter	2 h	CO3
11	Principal Component Analysis	Apply Principal Component Analysis (PCA) to Reduce Feature Dimensions using the Wine Dataset.	Google Colab/ Jupyter	2 h	CO3
MODULE IV · Exercises 12 : Probabilistic Graphical Models and Ensemble Learning (02 Hours)					
12	Bagging, Boosting	Implement and compare ensemble techniques: Bagging (Random Forest) Boosting (AdaBoost / Gradient Boosting)	Google Colab/ Jupyter	2 h	CO4
MODULE V · Exercises 13–15 : Machine Learning Model Validation, Optimization, and Regularization (06 Hours)					
13	**Model evaluation**	Evaluate Classification Models by Computing Accuracy, Precision, Recall, F1-Score, and Confusion Matrix using the Breast Cancer Dataset.	Google Colab/ Jupyter	2 h	CO5
14	Hyperparameter	Perform Hyperparameter Tuning using	Google Colab/	2 h	CO5

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
	Tuning	Grid Search with Cross-Validation on the Digits Dataset	Jupyter		
15	Machine Learning Model to Solve a Real-World Engineering	Design and Implement a Machine Learning Model to Solve a Real-World Engineering Problem using an Open Engineering Dataset (e.g., Energy Consumption, Traffic Flow, or Predictive Maintenance Dataset)	Google Colab/ Jupyter	2 h	**CO5**

****

Design studio- II: Innovation, Design Thinking & Intellectual Property
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs (15Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: This course aims to develop creative, innovative, and user-centric problem-solving abilities among students by introducing the principles of design thinking and innovation. It aims to equip learners with the skills to identify real-world problems, generate effective solutions, and develop functional prototypes using structured design methodologies. The course also aims to build awareness of intellectual property rights and foster collaborative, interdisciplinary thinking to prepare students as future-ready innovators capable of addressing complex challenges.

Prerequisites: - Basic Problem Solving, Creativity, Communication Skills

Course Learning Objectives:
This Course will enable students to:

1. Understand principles of design thinking and innovation processes.
2. Apply user-centric approaches to identify and solve real-world problems.
3. Develop creative prototypes using design tools and methodologies.
4. Gain knowledge of intellectual property rights (IPR) and innovation protection.
5. Build collaborative and interdisciplinary problem-solving skills.

Broad Course Description
This course introduces students to the principles and practices of innovation and design thinking, emphasizing a user-centric approach to solving real-world problems. It enables learners to understand the complete innovation lifecycle—from problem identification and ideation to prototyping and solution development—using structured design methodologies. Students will engage in hands-on activities to develop creative and practical prototypes, applying modern design tools and techniques. The course also provides foundational knowledge of Intellectual Property Rights (IPR), helping students understand how to protect innovations, manage ownership, and appreciate the legal and ethical aspects of creative work. By fostering collaboration and interdisciplinary teamwork, the course aims to build critical thinking, creativity, and problem-solving skills, preparing students to become innovative thinkers and future-ready professionals capable of addressing complex societal and technological challenges.

UNIT – I	**03 Hours**

Introduction to Design Thinking & Innovation: Design thinking process (Empathise, Define, Ideate, Prototype, Test), Innovation types, Case studies (product/service innovation). **Applications:** Solving campus and community problems.

AI Integration:
Use of AI tools for idea generation and user insight analysis. **Authentic Intelligence:** Understanding real user needs vs assumed needs.

UNIT – II	**03 Hours**

Empathy & Problem Definition: User research methods (interviews, observation), Persona creation, Journey mapping, Problem framing, Design challenges. **Applications:** Identifying real-world problems in healthcare, education, smart cities. **AI Integration:** AI-based sentiment analysis and user feedback interpretation. **Authentic Intelligence:** Ethical data collection and unbiased problem identification.

UNIT – III	**03 Hours**

Ideation & Concept Development: Brainstorming techniques, SCAMPER, Mind mapping, Concept selection, Feasibility analysis. **AI Applications:** Innovative product/service design. **AI Integration:** Generative AI for idea expansion and concept visualisation. **Authentic Intelligence:** Originality vs AI-generated ideas—critical evaluation.

UNIT – IV	03 Hours
Prototyping & Testing: Low-fidelity and high-fidelity prototypes, Wireframing, MVP development, User testing, Feedback loops. **Applications:** App prototypes, IoT concepts, service models. **AI Integration:** AI-assisted prototyping tools and UI/UX generators. **Authentic Intelligence:** Importance of human validation in AI-generated designs.	
UNIT – V	**03 Hours**
Intellectual Property & Innovation Protection: Types of IP (Patents, Copyrights, Trademarks, Trade secrets), Patent filing process, Prior art search, Technology transfer. **Applications:** Protecting student innovations and startup ideas. **AI Integration:** AI tools for patent search and novelty detection. **Authentic Intelligence:** Ethical innovation and avoiding plagiarism.	

RECOMMENDED TOOLS

- **Design Tools:** Figma, Canva, Adobe XD
- **Prototyping:** Balsamiq, Tinkercad
- **AI Tools:** ChatGPT, Ideation tools, UI generators
- **IP Tools:** Google Patents, WIPO databases

Laboratory / Studio Component

1. Identify a **real-world problem** through field observation.
2. Conduct **user interviews and create personas**.
3. Develop **problem statements and design challenges**.
4. Perform **ideation sessions (brainstorming, SCAMPER)**.
5. Create **low-fidelity prototypes (paper/wireframes)**.
6. Develop **digital prototypes using design tools**.
7. Conduct **user testing and collect feedback**.
8. Perform **basic patent search for the idea**.
9. Prepare **IP strategy for the solution**.
10. Final **design showcase + innovation pitch**.

Integrated Tools for Studio

- **Software:** Figma, Canva, Tinkercad
- **Platforms:** Google Forms (user feedback), Miro (brainstorming)
- **AI Tools:** Generative AI tools for ideation and prototyping
- **IP Platforms:** Google Patents, WIPO

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Apply design thinking methodologies to identify user needs and problem statements	L3
CO2	innovative solutions through ideation and prototyping	L3
CO3	Use modern tools for rapid prototyping and design validation	L3
CO4	Analyze user requirements and evaluate alternative design solutions	L4
CO5	Work collaboratively to design and present innovative solutions	L3

Table: Mapping Levels of COs to POs / PSOs													
COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	CO1	1	3	2	2	2	2	-	2	2	2	3	
CO2	CO2	1	2	3	2	3	2	-	2	3	2	3	
CO3	CO3	1	2	3	2	3	1	-	1	2	2	3	
CO4	CO4	1	1	2	1	2	2	1	3	1	2	3	
CO5	CO5	1	2	3	1	2	2	-	2	3	3	3	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Change by Design – Tim Brown
   Covers fundamentals of design thinking, innovation strategies, and human-centered design (CO1, CO2, CO3).
2. Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days – Jake Knapp*Focuses on rapid prototyping, ideation, and solution validation*
3. WIPO Intellectual Property Handbook – World Intellectual Property Organization *Provides comprehensive understanding of IPR concepts, laws, and protection mechanisms (CO1, CO5).*

REFERENCE BOOKS:

1. IDEO Design Thinking Resources
   Practical toolkits, case studies, and real-world Applications of design thinking (CO2, CO3, CO4).
2. Stanford school Materials
   Structured methodologies, design frameworks, and collaborative innovation practices
3. The Design of Everyday Things – Don Norman
   Enhances understanding of user experience and usability principles (CO2, CO3).
4. Creative Confidence – Tom Kelley, David Kelley

E-RESOURCES:

1. World Intellectual Property Organization – Official Learning Resources
   https://www.wipo.int
   (IPR tutorials, case studies, global standards
2. IDEO U – Design Thinking Courses
   https://www.ideou.com
   (Hands-on innovation and design thinking modules
3. Stanford University – d.school Open Resources
   https://dschool.stanford.edu/resources
   (Design thinking toolkits and guides
4. Coursera / edX
   (Courses on innovation, design thinking, and IPR
5. Google Patents

https://patents.google.com
(Patent search and analysis for practical IPR understanding
Evaluation Scheme (2-Credit Studio)

Internal (100 Marks – No End Semester Exam

Recommended) Component Marks

Continuous Assessment (Activities) 30

Design Challenge / Assignments	20
Prototype Development	20
IP Report / Patent Search	10
Final Project + Presentation	20

****

Introduction to Data Science

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs L
L–T–P-J	:	2-0-0-0			

Course Vision: This course aims to equip undergraduate students with strong foundational and practical knowledge in data science, enabling them to effectively collect, preprocess, analyze, and visualize data using modern tools and techniques. It aims to develop analytical thinking and problem-solving skills to derive meaningful insights from data, while fostering the ability to apply statistical methods, machine learning concepts, and scalable data analytics approaches for real-world, data-driven decision-making.

Prerequisites: - Python, Probability and Statistics

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamental data science concepts and workflows
2. Apply statistical techniques and visualization for data analysis
3. Perform data cleaning and preprocessing
4. Use feature selection and dimensionality reduction methods
5. Gain exposure to scalable data analytics techniques

Broad Course Description: This course provides a comprehensive introduction to data science, focusing on the complete data lifecycle including data collection, cleaning, preprocessing, analysis, visualization, and interpretation. Students learn to apply statistical methods, feature engineering, and dimensionality reduction techniques using Python-based tools such as Pandas, NumPy, and Scikit-learn. The course emphasizes exploratory data analysis, data wrangling, and predictive modeling, along with practical visualization techniques for meaningful insight generation. It also introduces scalable and accelerated data analytics concepts using modern GPU-based tools, enabling efficient handling of large datasets. Through hands-on experiments and real-world case studies, students develop the ability to transform raw data into actionable insights for informed decision-making in data-driven environments.

UNIT – I	06 Hours

Basic Concepts and Python Packages: Predictive Modelling, Introduction: Data Preparation, Importance of Data Preparation, Data Cleaning, Feature Selection, Data Transformation, Python Packages: NumPy, Matplotlib, Pandas, SciPy, Scikit-learn, DataFrames, Loading Machine Learning Data. **Applications:** Data preprocessing in real-world problems**,** Basic analytics workflows.

AI Integration: Data preparation for AI systems. **Authentic Intelligence:**Understanding data quality and reliability.

UNIT – II	06 Hours

Descriptive Statistics and Data Visualization: Sampling, Data Analysis Process, Mean, Median, Standard Deviation, Skewness, Kurtosis, Data Visualization Techniques: Box Plot, Scatter Plot, Histogram, Bar Chart, Pie Chart, Heat Map, Correlation Analysis, Pivot Tables, ANOVA, Data Preprocessing, Rescaling, Standardization, Normalization, Binarization, Univariate and Bivariate Analysis, Recursive Feature Elimination, Principal Component Analysis. **Applications:** Data analysis and reporting Visualization-driven decision making. **AI Integration:** Statistical modeling and visualization in AI. **Authentic Intelligence:** Interpreting statistical and visual insights correctly

UNIT – III	06 Hours

Data Cleaning and Feature Selection: Handling Missing Data, Outlier Detection and Removal, Statistical Imputation, KNN Imputation, Iterative Imputation, Feature Selection Techniques, Methods for Categorical and Numerical Data, Feature Selection for Outputs, Recursive Feature Elimination,

Importance of Feature Selection. **AI Applications:** Data cleaning in industry datasets, Feature engineering workflows. **AI Integration:** Improving model performance through feature selection. **Authentic Intelligence:** Ensuring reliable and unbiased data.	
UNIT – IV	**03 Hours**
Data Transformation and Dimensionality Reduction: Scaling Techniques, Min-Max Scaling, Standard Scaling, Handling Outliers, Encoding Categorical Data, Transforming Data Distributions, Feature Engineering, PCA, LDA, SVD Techniques **Applications:** Data transformation pipelines,High-dimensional data analysis. **AI Integration:** Feature optimization for machine learning. **Authentic Intelligence:** Reducing complexity while preserving information.	
UNIT – V	**03 Hours**
Accelerated Data Analytics (NVIDIA DLI): CPU vs GPU in Data Processing, Need for Fast Data Analytics, Introduction to RAPIDS, GPU-Accelerated Data Workflows, Handling Large Datasets Efficiently, Basic Concept of Accelerated DataFrame Operations, Overview of Machine Learning Acceleration, Identifying Processing Bottlenecks, Real-World Applications of Accelerated Data Analytics. **Case Study (NVIDIA DGX):** Population-scale data analysis for epidemic control, Integration of multiple datasets, Real-time data analysis and iterative model updates, Decision-making using large-scale data insights**. Applications:**Large-scale data analysis,Real-time decision-making systems. **AI Integration:** GPU-accelerated ML pipelines, High-performance AI systems. **Authentic Intelligence:** Efficient handling of large datasets,Scalable and reliable data-driven solutions.	

Supporting Experiments

1. Develop a Python program to read different types of datasets (.txt, .csv, Excel, XML, JSON) from web and local disk using Pandas, merge them based on a common attribute (e.g., employee), and store the merged dataset into a new CSV file in a specified location.
2. Write a Python program to perform Exploratory Data Analysis (EDA) on a given dataset and generate visualizations including box plot, scatter plot, histogram, bar chart, pie chart, and heatmap to analyze data distribution and relationships.
3. Write a Python program to handle missing values in a dataset using Statistical Imputation, KNN Imputation, and Iterative Imputation methods and compare their effectiveness.
4. Write a Python program to detect and remove outliers from a dataset using Standard Deviation method, Interquartile Range (IQR) method, and automatic outlier detection techniques, and compare the results using appropriate plots.Case Study: Accelerated Data Processing using Pandas and RAPIDS

Problem Statement:
In real-world data science Applications, large datasets often lead to increased processing time and reduced efficiency. This case study explores how accelerated data processing techniques can improve performance by comparing traditional Pandas operations with GPU-accelerated alternatives such as RAPIDS cuDF.

Dataset:
A moderately large CSV dataset (e.g., sales data, customer transactions, or sensor data) containing multiple columns and thousands/millions of records.

Tasks:
1. Load the dataset using Pandas.
2. Perform basic data operations such as:
   · Filtering rows based on conditions
   · Aggregation (groupby, sum, mean)

· Sorting data based on a column

3. Measure execution time for each operation using Python's time module.
4. (Optional) Perform the same operations using RAPIDS cuDF
5. Compare the execution time between Pandas (CPU) and cuDF (GPU).

Course Outcome	**Description**
CO1	Explain basic data science concepts and data preprocessing techniques
CO2	Apply statistical methods and visualization techniques for data analysis
CO3	Perform data cleaning and feature selection for efficient data preparation
CO4	Apply data transformation and dimensionality reduction techniques
CO5	Understand scalable data analytics concepts using modern tools

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2		1	2	-	-	-	-	1	-	2	
CO2	2	3	2	3	2	-	-	-	-	1	-	3	
CO3	2	3	2	2	3	-	-	-	-	1	-	3	
CO4	2	2	3	2	3	-	-	1	-	1	-	3	
CO5	2	2	2	3	3	-	-	-	-	1	-	3	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

References

1. Joel Grus – *Data Science from Scratch*
2. Wes McKinney – *Python for Data Analysis*
3. Jake VanderPlas – *Python Data Science Handbook*
4. Online documentation (Pandas, Scikit-learn, Matplotlib)
5. DLI Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-DS-03+V1#:~:text=The%20course%20prerequisites%20include:%20*%20Basic%20understanding,the%20NVIDIA%20Deep%20Learning%20Institute%20at%20www.nvidia.com/dli
6. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-DS-04+V1#:~:text=Build%20and%20compare%20classification%20models,learning%20 and%20clustering%20on%20embeddings.

Evaluation Scheme (2-Credit Studio)
Internal (100 Marks – No End Semester Exam

Recommended) Component Marks

Continuous Assessment (Activities) 30

Design Challenge / Assignments	20
Prototype Development	20
IP Report / Patent Search	10
Final Project + Presentation	20

****

Object Oriented Programming Using Java

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs (15Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: The course aims to develop strong foundations in Java programming and object-oriented design for building robust, scalable, and platform-independent software applications. It enables students to understand core programming concepts such as classes, objects, inheritance, interfaces, exception handling, and multithreading for effective software development. The course also equips learners with practical skills in using Java libraries, collections, file handling, and JDBC for database connectivity to develop real-world applications and enterprise-level systems.

Prerequisites: NIL

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamental principles of object-oriented programming using Java.
2. Develop Java Applications using classes, objects, arrays, and strings.
3. Implement inheritance, interfaces, and handle exceptions effectively.
4. Apply multithreading and perform I/O operations in Java.
5. Use Java Collections framework and perform database connectivity using JDBC.

Broad Course Description

Java is a high-level, object-oriented programming language developed by Sun Microsystems in 1995 and later acquired by Oracle Corporation. It was designed with the principle of "write once, run anywhere," meaning that Java programs can run on any system equipped with a Java Virtual Machine (JVM). This platform independence, along with its simplicity and robustness, has made Java one of the most widely used programming languages in the world. Java follows object-oriented principles such as encapsulation, inheritance, and polymorphism, which help developers build modular, reusable, and maintainable code.
One of Java's key strengths is its strong memory management and security features. It uses automatic garbage collection to manage memory efficiently, reducing the risk of memory leaks. Java also provides a rich standard library (Java API) that supports a wide range of functionalities, including data structures, networking, file handling, and graphical user interfaces. Additionally, Java is known for its reliability and portability, making it a preferred choice for building enterprise-level Applications, web services, and large-scale systems

UNIT – I	03 Hours

Introduction to Java:

Features of OOP, Characteristics/Buzz words of Java, Java Environment: JDK, JVM, JRE, Fundamental Programming Structure in Java, Variables, Data Types, Operators & Expressions, Control Statements, Iteration Statements, Command Line Arguments. Single and Multidimensional Arrays.
(Textbook 1: Chapter:1-5)

UNIT – II	03 Hours

Classes and Objects:

Classes & Objects: Defining Classes & Objects, Access Specifies, Constructors, Overloading Constructor, Method Overloading, Passing and Returning object form Method, new operator, finalize() method, this keyword, Static Keyword, Encapsulation, Polymorphism, Strings.
(Textbook 1: Chapter: 6)

UNIT – III	03 Hours

Inheritance:

Inheritance: Defining a Inheritance, Types of Inheritance, Constructor in subclass, Method Overriding, super keyword, abstract keyword, final keyword. **Interfaces:** Defining a Interface, Implementing a Interface, Difference between Interface & Classes, Extending a Interface. (Textbook 1: Chapter: 8, 9)	
UNIT – IV	**03 Hours**
Package & Exceptions: Packages: Usage of Package, Classpath, Importing a Package. Exceptions: Definition of Exception, Classification of Exception, Structure of Try & catch block, Error Vs Exception, Throw Keyword, Throws Keyword, Finally Keyword, Custom Exception.. (Textbook 1: Chapter: 9, 10)	
UNIT – V	**03 Hours**
Multi Threading: Multithreading: Multi-Threaded Programming: What are threads? How to make the classes threadable? Extending threads, Implementing runnable, Synchronization, Thread priorities. (Textbook 1: Chapter: 11)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)AI-Assisted Code Generation & Explanation: Use AI tools (e.g., GitHub Copilot, ChatGPT) to generate Java class structures, methods, and OOP examples, helping students understand concepts like inheritance and polymorphism interactively.

- **Automated Code Review & Feedback:** AI-driven platforms analyze Java programs for correctness, efficiency, and adherence to OOP principles (encapsulation, abstraction), providing instant feedback and suggestions.
- **Code Visualization & Concept Mapping:** AI-powered visualization tools represent object interactions, class hierarchies, and memory allocation, making abstract OOP concepts easier to grasp..
- **Critical Evaluation of AI-Generated Code:** Students analyze AI-generated Java code for correctness, efficiency, and design quality, fostering deeper understanding rather than passive acceptance.
- **Ethical Use of AI in Programming:** Emphasize responsible use of AI tools, avoiding over-reliance and ensuring originality, especially in assignments and projects.

Applications:

1. Enterprise Applications
2. Web Applications
3. Desktop GUI Applications
4. Financial Applications.

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

1. Write a program to implement a simple ATM system where:
   - A menu-driven interface is provided using switch statement
   - The user can withdraw, deposit, or check balance
   - Use while loop to keep the session active until the user exits
2. Develop a Java program that accepts employee details (name, age, department) as command line arguments and displays them in a formatted output. Validate inputs (e.g., age must be numeric and > 18).
3. Design a class Student with private fields: name, rollNo, and marks.
   - Use constructor overloading to allow both default and parameterized object creation.
   - Apply encapsulation using getters and setters.Use the this keyword to resolve variable shadowing.
   - Track total students using a static variable and method.
   - Create another method that accepts a Student object as parameter and returns the same object with bonus marks added.
4. Create an abstract base class Shape with an abstract method area().
   - Derive classes Circle, Rectangle, and Triangle that override area() method using runtime polymorphism.
   - Demonstrate calling overridden methods using a base class reference.
5. Create a class \`Employee\` with attributes \`empId\`, \`name\`, \`salary\`. Create a subclass \`Manager\` that adds \`department\` and overrides \`display()\` method. Demonstrate access to base and derived class members.
6. Create a base class \`Vehicle\` with method \`showSpeed()\`. Derive \`Car\` and \`Bike\` classes that override this method. Demonstrate dynamic dispatch using base class reference.
7. Create an interface PersonDetails with method display(). Extend it in another interface StaffDetails with method calculateSalary().
   - Implement StaffDetails in a class Professor.
   - Simulate error scenarios like null values or negative salary using throw and throws keywords.
   - Use a package university.staff and demonstrate use of classpath and import statements in a driver class.
8. Design a package student.registration with a class Student and interface Registrable.
   - The interface should declare a method register().
   - Implement the interface and throw a custom exception InvalidRegistrationException if age is below 18.
   - Use try-catch block and a finally block to confirm registration closure.
9. Design a class TicketCounter where multiple users (threads) try to book tickets simultaneously.
   - Use thread synchronization to prevent race conditions.
   - Create user threads by both extending Thread and implementing Runnable.
   - Assign thread priorities based on user type (e.g., VIP, Regular).
   - Use an enum UserType { VIP, REGULAR } to distinguish users and use valueOf() to convert string input.
10. Create a class BankAccount that supports deposit and withdrawal.
    - Spawn multiple threads to simulate transactions concurrently using Runnable.
    - Ensure thread synchronization for consistency.

- Use enum TransactionType { DEPOSIT, WITHDRAW } and demonstrate values() and valueOf() methods.
- Show how thread priority affects execution order (optional based on thread scheduler).

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the **real software engineering problems**. **Project 1 – Online Banking SimulationProblem:** Simulate basic banking operations like deposit, withdrawal, and balance check.

- **Concepts used:** OOP (Inheritance, Polymorphism), Exception Handling, Methods.
- **Industry relevance:** Forms the foundation of real-world banking and fintech systems.

Project 2 – Library Management System

- **Problem:** Track books, issue/return records, and user details..
- **Concepts used:** Classes, Collections (List/Map), File Handling, Encapsulation.
- **Industry relevance:** Used in libraries, universities, and digital catalog systems.

Project 3 – Quiz Application

- **Problem:** Conduct quizzes with scoring and result evaluation.
- **Concepts used:** Arrays, Control Statements, Classes, String Handling.
- **Industry relevance:** Applied in e-learning and online assessment platforms.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Understand object oriented programming concepts and basics of JAVA to solve simple problems.	L2
CO2	Construct a class involving data members and methods for the given scenario.	L3
CO3	Apply the concepts of inheritance to implement Java Applications	L3
CO4	Apply the concepts of packages, interfaces and exception handling.	L3
CO5	Develop Java Applications using multithreading.	L3

Table: Mapping Levels of COs to POs / PSOs												
COs	**Program Outcomes (POs)**										**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	3	2						2	3	2
CO2	3	3	2	3	3						2	3	2
CO3	3	3	2	3	3						2	3	2
CO4	3	3	2	3	3						2	3	2

CO5	2	3	2	3	3						2	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Suggested Tools:

- Programming Tools: NetBeans/Eclipse/Visual Studio

TEXT BOOKS:

1. The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, - TMH.
2. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
3. Programming with Java A Primer – E.Balaguruswamy,Mc Grawhill.

REFERENCE BOOKS:

1. Core Java Volume-I Fundamentals Horstmann & Cornell, - Pearson Education. - Eight Edition.
2. Head First Java: A Brain-Friendly Guide, 2nd Edition- Kathy Sierra, Bert Bates.

E-RESOURCES:

- Coursera
- https://www.geeksforgeeks.org/java/java/
- Nptel: Java

Activity Based Learning (Suggested Activities in Class)

- Object Interaction Role Play: Students act as objects and simulate method calls and message passing between objects.
- Debugging Challenge: Students identify and fix errors in faulty Java OOP programs.
- Mini Project Development: Students build small Applications (e.g., Library System, Banking System).

****

DevOps Fundamentals & Product Launch

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	04 Hours	**Total Hours**	:	30 Hrs (15HrsL +15Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: The course aims to develop strong foundations in DevOps practices and cloud-native software delivery for designing scalable, automated, and reliable software systems. It enables students to understand CI/CD pipelines, version control workflows, containerization, orchestration, and infrastructure automation using modern tools and frameworks. The course also equips learners with practical skills in deploying, monitoring, and securing applications through DevSecOps practices, enabling efficient collaboration between development and operations teams in real-world production environments.

Course Learning Objectives:

This Course will enable students to:

1. Understand the DevOps philosophy, culture, and its role in modern software delivery pipelines.
2. Learn version control best practices, CI/CD pipeline design, and build automation strategies.
3. Master containerization with Docker and container orchestration using Kubernetes.
4. Implement Infrastructure as Code, observability, and cloud deployment using industry-standard tools.

Broad Course Description

DevOps breaks down silos between development and operations, enabling faster, more reliable software delivery. This course covers the complete DevOps lifecycle: Git-based version control with GitFlow and trunk-based development, Jenkins and GitHub Actions CI/CD pipelines, automated testing strategies, Docker containerization with multi-stage builds and Docker Compose, Kubernetes orchestration (Pods, Services, Deployments, ConfigMaps, Helm, HPA), Terraform and Ansible Infrastructure as Code, AWS cloud basics (EC2, S3, RDS, Lambda), and Prometheus/Grafana/ELK observability stack. DevSecOps principles including SAST, DAST, container vulnerability scanning with Trivy, and secrets management with HashiCorp Vault are integrated throughout. Students build complete CI/CD pipelines and production deployment systems applicable to real-world engineering roles.

UNIT – I	03 Hours

DevOps Foundations & Version Control: DevOps philosophy – CAMS model (Culture, Automation, Measurement, Sharing), Lifecycle phases, Agile and DevOps alignment, DORA metrics. Git fundamentals – init, add, commit, push, pull, branching and merging. GitHub/GitLab – remote repositories, pull requests, code reviews. Git workflows – GitFlow, GitHub Flow, Trunk-Based Development. (Ref: Gene Kim, The DevOps Handbook Ch. 1–4)

Textbook Reference:

• Gene Kim et al., The DevOps Handbook, 2nd Ed – Ch. 1–4 (IT Revolution, 2021)

• Scott Chacon & Ben Straub, Pro Git – Ch. 1–3 (Apress; free at git-scm.com)

UNIT – II	03 Hours

CI/CD Pipelines & Automation: CI concepts – build automation, automated testing, code quality gates. Jenkins – installation, declarative pipeline syntax. GitHub Actions – workflows, triggers, action marketplace. Automated testing in CI – unit tests, integration tests, code coverage reports. Artifact management – Docker Registry, JFrog. Blue-Green and Canary deployment strategies. (Ref: Jez Humble, Continuous Delivery Ch. 1–5)

Textbook Reference:

• Jez Humble & David Farley, Continuous Delivery – Ch. 1–5 (Addison-Wesley, 2010)

• GitHub Actions Official Documentation (docs.github.com/en/actions)

UNIT – III	03 Hours

Containerization with Docker: Virtualization vs. Containerization. Docker architecture – Docker Engine, Images, Containers, Registry. Dockerfile – writing, layering, multi-stage builds. Docker CLI –

build, run, stop, exec, logs. Docker Compose – multi-container Applications, networking, and volumes. Container networking modes. Docker Hub and private registries. Security scanning and best practices. (Ref: Nigel Poulton, Docker Deep Dive Ch. 1–8)
Textbook Reference:
• Nigel Poulton, Docker Deep Dive – Ch. 1–8 (Self-Published, 2023)
• Docker Official Documentation (docs.docker.com)

UNIT – IV	03 Hours

Container Orchestration with Kubernetes: Kubernetes architecture – Master/Worker nodes, Pods, Services, Deployments, ReplicaSets. ConfigMaps and Secrets, Namespaces, kubectl CLI. Kubernetes networking – ClusterIP, NodePort, LoadBalancer, Ingress. Persistent Volumes and PVCs. Helm package manager. Horizontal Pod Autoscaler (HPA). Managed Kubernetes – EKS, GKE, AKS. (Ref: Brendan Burns, Kubernetes: Up and Running Ch. 1–8)
Textbook Reference:
• Brendan Burns et al., Kubernetes: Up and Running, 3rd Ed – Ch. 1–8 (O'Reilly, 2022)
• Kubernetes Official Documentation (kubernetes.io/docs)

UNIT – V	03 Hours

IaC, Monitoring & DevSecOps: Terraform – providers, resources, state management, modules. Ansible – playbooks, roles, inventory management. AWS basics – EC2, S3, RDS, Lambda. Monitoring and observability – Prometheus, Grafana dashboards, ELK Stack (Elasticsearch, Logstash, Kibana).
Alerting and on-call practices. DevSecOps – SAST, DAST, Trivy container scanning, Vault secrets management. SRE principles – SLIs, SLOs, SLAs. (Ref: Kief Morris, Infrastructure as Code Ch. 1–6)
Textbook Reference:
Kief Morris, Infrastructure as Code, 2nd Ed – Ch. 1–6 (O'Reilly, 2020)
Betsy Beyer et al., Site Reliability Engineering – Ch. 1–4 (O'Reilly, 2016)

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Unit 3	**An Even Easier Introduction to CUDA**	Accelerated Computing	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-v1:DLI+T-AC-01+V1	CUDA introduces GPU-accelerated parallel computing — directly relevant to understanding Docker GPU containers (NVIDIA Container Toolkit) and GPU-enabled containerized AI workloads in Unit 3.
	AI Infrastructure and Operations Fundamentals	Infrastructure / AI Ops			learn.nvidia.com/courses/course-detail?course_id= course-	Essential 8-hr free course on GPU cluster management, DGX administration,

Units 4–5			8 Hrs	**Yes**	v1:DLI+S-IN-01+V1	NCCL, and AI Enterprise — directly maps to Kubernetes-based GPU orchestration, IaC, and monitoring in Units 4–5. Certificate awarded.
Unit 5	**Accelerate Data Science Workflows With Zero Code Changes**	Data Science / RAPIDS	2 Hrs	—	courses.nvidia .com/ course-v1:DLI+T-DS-03+V1	GPU-accelerated data science pipelines via RAPIDS — reinforces Unit 5 cloud deployment and monitoring of data-intensive AI microservices deployed in GPU-enabled Kubernetes clusters.
Unit 5	**Exploring Adversarial Machine Learning**	Deep Learning / Security	2 Hrs	—	learn.nvidia.com/courses/ course-detail?course_id= course-v1:DLI+T-FX-06+V1	Adversarial ML covers AI-based threat modeling — maps directly to DevSecOps (SAST, DAST, AI-powered security pipelines) and security hardening topics in Unit 5.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)AI-Assisted Pipeline Generation: Use GitHub Copilot to generate CI/CD YAML configurations; evaluate correctness, security, and edge-case handling.

- **Intelligent Monitoring Analysis:** Explore AI-based anomaly detection in Prometheus/Grafana; understand the difference between rule-based and ML-based alerting.
- **Reflective Automation Thinking:** Identify pipeline stages suitable for full automation versus those requiring human approval gates and judgment.
- **AI in DevSecOps:** Evaluate AI-powered security scanning tools (SAST, vulnerability detection) for reliability, false positive rates, and actionability.
- **Infrastructure Cost Optimization with AI:** Evaluate AI recommendations for cloud resource sizing, reserved instance planning, and cost-aware engineering.

Applications:

- Continuous Software Delivery in Enterprise Product Companies
- Cloud-Native Application Deployment and Scaling
- Microservices Infrastructure Management
- Automated Testing and Quality Assurance Pipelines
- Site Reliability Engineering (SRE) at Scale

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Set up a Git repository; practice branching strategies, pull requests, and structured code reviews.
- Build a Jenkins pipeline to automate build, test, and artifact archiving for a Node.js application.
- Create a GitHub Actions CI/CD workflow that runs tests and deploys to a staging environment.
- Write a multi-stage Dockerfile for Node.js/Python; build, tag, and push to Docker Hub.
- Deploy a multi-container application using Docker Compose (frontend + backend + database).
- Set up local Kubernetes cluster with Minikube; deploy, scale, and expose a containerized application.
- Implement Kubernetes rolling updates, rollbacks, and Horizontal Pod Autoscaler.
- Write Terraform config to provision an AWS EC2 instance with security groups and S3 bucket.
- Set up Prometheus and Grafana to monitor a containerized application with custom alerting rules.
- Implement a DevSecOps pipeline with SonarQube (code quality), Trivy (container scanning), and Vault (secrets).

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – Full CI/CD Pipeline for Microservices App

Problem: Complete CI/CD pipeline for a 3-tier microservices application with automated testing and Kubernetes deployment.

Concepts used: GitHub Actions, Docker, Kubernetes, Helm

Industry relevance: Standard DevOps workflow in product-based companies.

Project 2 – Infrastructure Automation with IaC

Problem: Provision cloud infrastructure (VPC, EC2, RDS, Load Balancer) using Terraform and configure with Ansible.

Concepts used: Terraform, Ansible, AWS, IaC principles

Industry relevance: Cloud and platform engineering teams.

Project 3 – Observability Platform Setup

Problem: Build a complete monitoring and logging platform for a distributed application.

Concepts used: Prometheus, Grafana, ELK Stack, Alerting

Industry relevance: SRE and platform engineering roles.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand DevOps philosophy and apply Git version control best practices in team-based development workflows.	L2
2	Design and implement CI/CD pipelines using Jenkins and GitHub Actions for automated build, test, and deployment.	L3
3	Containerize Applications using Docker and manage multi-container deployments with Docker Compose.	L3
4	Deploy, scale, and manage containerized Applications using Kubernetes with networking and storage configurations.	L4
5	Implement IaC using Terraform/Ansible, establish observability, and apply DevSecOps security practices.	L5

Table: Mapping Levels of COs to POs / PSOs

	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	3	2						2	3	2
CO2	3	3	3	3	3						2	3	2
CO3	3	3	2	3	3						2	3	2
CO4	3	3	3	3	3						3	3	3
CO5	3	3	3	3	3						3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools:

- Git, GitHub / GitLab
- Jenkins, GitHub Actions
- Docker, Docker Compose
- Kubernetes, Helm
- Terraform, Ansible
- Prometheus, Grafana, ELK Stack

TEXT BOOKS:

1. Gene Kim et al., The DevOps Handbook, 2nd Ed, IT Revolution Press, 2021.
2. Jez Humble & David Farley, Continuous Delivery, Addison-Wesley, 2010.
3. Brendan Burns et al., Kubernetes: Up and Running, 3rd Ed, O'Reilly, 2022.REFERENCE BOOKS:
4. Kief Morris, Infrastructure as Code, 2nd Ed, O'Reilly, 2020.
5. Betsy Beyer et al., Site Reliability Engineering, O'Reilly, 2016.

E-RESOURCES:

- https://kodekloud.com
- https://training.linuxfoundation.org
- https://www.coursera.org/learn/devops-on-aws
- https://learn.nvidia.com (NVIDIA DLI Free Courses)

Activity Based Learning (Suggested Activities in Class)

- DevOps simulation: build-break-fix pipeline exercises in Minikube sandbox environments.
- Weekly retrospectives simulating Agile stand-up meetings and sprint reviews.
- Guest lectures from DevOps engineers and Site Reliability Engineers at product companies.
- Incident response role-play: teams diagnose and resolve a simulated production outage.

****

R Programming

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs (15 Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision:

The course aims to develop strong foundations in R programming for data analysis, statistical computing, and machine learning applications. It enables students to understand R programming constructs, data structures, visualization techniques, and statistical methods for meaningful data interpretation. The course also equips learners with practical skills to implement hypothesis testing, predictive models, and basic machine learning algorithms using R for real-world data-driven decision-making and analytical problem solving.

Prerequisites: Python Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamentals and environment of R programming
2. Apply programming constructs and control structures in R
3. Develop graphical visualizations using R
4. Perform statistical analysis and hypothesis testing
5. Implement basic machine learning models using R

Broad Course Description

The course R Programming provides a comprehensive foundation in R programming with emphasis on data structures, programming constructs, visualization, statistical analysis, and machine learning techniques. Students will learn to handle various data types and structures, develop efficient R programs using control statements and functions, and generate insightful visualizations. The course also introduces statistical techniques such as probability distributions, hypothesis testing, and ANOVA. Further, it covers implementation of machine learning models including regression, decision trees, and random forests. The course equips learners with practical skills for real-world data analysis and intelligent decision-making.

UNIT – I	**05 Hours**

Introduction to R

How to run R, R sessions and Functions, Basic Maths, Variables, Data Types, Vectors, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes.

(Text Book-1: Chapter 1: 1.1 to 1.4, 1.6 Chapter 2: 2.1,2.2,2.4)

UNIT – II	**06 Hours**

R Programming Structure

Control Statements, Loops – Looping Over Non-vector sets, If-else, Arithmetic and Boolean operators, Default value for argument, Return values, Function with No pointers in R, Recursion, Sorting and Searching

(Text Book-1: Chapter 3: 3.1 to 3.3)

UNIT – III	**07 Hours**

Plots and Graphs

Graphics, Creating Graphs, The workhorse of R Base Graphics, the plot () Function, Customising Graphs, Saving Graphs to Files

(Text Book 2: Chapter 8: 8.1 to 8.8)

UNIT – IV	**06 Hours**

T Testing

Probability Distributions, Normal Distribution – Binomial Distribution – Poisson Distributions, Basic

Statistics, Correlation and Covariance, T-Tests, ANOVA (Text Book2: Chapter 10: 10.1 to 10.6)	
UNIT – V	**06 Hours**
ML Model Implementation Linear Models, Simple Linear Regression and Multiple Regression, Generalized Linear Models, Nonlinear Models, Splines-Decision-Random Forests (Text Book2: Chapter 12: 12.1 to 12.9)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Programming**: Utilize AI tools (ChatGPT, Copilot) to assist in writing, debugging, and optimizing R code.
- **Data-Driven Intelligence**: Apply R for analyzing real-world datasets to extract meaningful insights.
- **Automated Data Visualization**: Use intelligent tools to generate and refine visual analytics.
- **AI-Augmented Statistical Modeling**: Integrate AI suggestions in building predictive models and interpreting results.
- **Hybrid AI–Data Analytics Approach**: Combine R programming with AI tools for efficient data analysis workflows.
- **Responsible Data Analysis**: Ensure ethical handling of data, transparency, and reproducibility in analysis.

AI Tools:

- Rstudio
- GitHub Copilot
- Kaggle
- Jupyter Notebook
- Google Colab

Applications:

1. Data analytics and visualization
2. Business intelligence and reporting
3. Financial modeling and forecasting
4. Healthcare data analysis
5. Predictive analytics using machine learning
6. Research and academic data analysis

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation using R and real-world datasets.

Lab Experiments

1. Installation and working with R environment
2. Programs using variables, data types, and vectors
3. Implementation of control statements and loops
4. Writing user-defined functions and recursion
5. Working with data frames, matrices, and lists

6. Creating and customizing plots using plot()
7. Saving graphical outputs
8. Statistical analysis (mean, variance, correlation)
9. Hypothesis testing (t-test, ANOVA)
10. Implementation of regression models
11. Implementation of decision trees and random forests

Mini Project Component

Industry-Driven Projects (IBM / Google Quantum Style)

These projects simulate real-world quantum AI Applications and research problems.

Project 1 – Exploratory Data Analysis (EDA)

- **Problem**: Analyze a real-world dataset and extract insights
- **Concepts used**: Data cleaning, visualization, summary statistics
- **Industry relevance**: Business analytics, healthcare

Project 2 – Statistical Modeling using R

- **Problem:** Build a regression or classification model
- **Concepts used:** Correlation, regression, hypothesis testing
- **Industry relevance:** Finance, prediction systems

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain R programming concepts and data structures	L2
2	Apply control structures and functions in R	L3
3	Analyze and visualize data using graphs	L4
4	Evaluate statistical methods and hypothesis tests	L5
5	Develop machine learning models using R	L6

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2			2							2	2
CO2	3	3	2	2	3							3	2
CO3	2	3	2	2	3							3	2
CO4		2			2	3						2	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. The Art of R Programming, Norman Matloff, cengage Learning: Efficient R Programming: A Practical Guide to Smarter Programming 1st Edition, Colin Gillespie & Robin Lovelace –First Edition
2. Jared P Lander, R for Everyone: Advanced Analytics and Graphics, Second edition 2017.

REFERENCE BOOKS:

1. R cookbook, Paul Teetor, Oreilly: R cookbook [R CKBK] paperback
2. R in Action, Rob kabacoff, Manning: Data Analysis and Graphics with RNov 5, 2018

E-RESOURCES:

- https://cran.r-project.org
- https://www.rstudio.com
- https://www.kaggle.com
- https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Data visualization challenge using real

dataset Activity 2: Statistical analysis case study

Activity 3: Mini hackathon on data analytics using R

Activity 4: AI-assisted coding and debugging

exercise

****

Introduction to Robotics

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs (15 Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0–2			

Course Vision: The course aims to develop strong foundations in robotics and robotic system design for analyzing, modeling, and controlling robotic manipulators used in real-world applications. It enables students to understand robot anatomy, coordinate transformations, kinematics, dynamics, and control techniques for precise motion and task execution. The course also equips learners with practical skills in trajectory planning, Jacobian-based analysis, and simulation-based implementation for designing intelligent robotic systems in industrial automation, manufacturing, and modern engineering applications.

Prerequisites: Basic Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. To introduce the fundamentals of robotics, including robot anatomy, configurations, and degrees of freedom.
2. To develop an understanding of coordinate transformations and rotation matrices used in robotic systems.
3. To enable modeling of robotic manipulators using forward kinematics and Denavit-Hartenberg (D-H) parameters.
4. To equip students with techniques to solve inverse kinematics and perform velocity and static analysis using Jacobians.
5. To provide knowledge of dynamic modeling, trajectory planning, and control strategies for robotic manipulators.

Broad Course Description

Introduction to Robotics is a skill-based course, it introduces the fundamental concepts of robotics, including robot anatomy, configurations, and degrees of freedom. It covers coordinate transformations and rotation matrices essential for describing robotic motion. The course focuses on kinematic modeling of manipulators using Denavit-Hartenberg (D-H) parameters and transformation matrices.

Students will learn forward and inverse kinematics for position and motion analysis. Velocity and static analysis using Jacobian methods are also discussed. The course further includes dynamic modeling using Lagrange-Euler and Newton-Euler formulations. Trajectory planning techniques in joint and Cartesian spaces are explored. Control strategies such as PID control, computed torque control, and force control are introduced. A practical lab component supports simulation, modeling, and control of robotic manipulators.

UNIT – I	03 Hours

FUNDAMENTALS OF ROBOTICS:

Evolution of robots and robotics, Robot anatomy: links, joints, degrees of freedom (DOF), Arm configurations (Cartesian, Cylindrical, Spherical, Articulated) and Wrist configurations and end-effectors.(Text Book-1: Chapter 1)

UNIT – II	03 Hours

COORDINATE TRANSFORMATIONS:

Mapping between rotated and translated frames, Combined rotation and translation of vectors and Fundamental rotation matrices (X, Y, Z axes) (Text Book1: Chapter 2)

UNIT – III	03 Hours

KINEMATICS OF MANIPULATORS:

Kinematic modeling of manipulators, Denavit-Hartenberg (D-H) notation, Transformation between

adjacent links and Manipulator transformation matrix (Text Book1: Chapter 3)	
UNIT – IV	**03 Hours**
INVERSE KINEMATICS AND VELOCITY ANALYSIS: Inverse kinematics and its solvability, Analytical and numerical solution techniques, Linear and angular velocity of rigid bodies, Jacobian matrix and velocity propagation and Static analysis of manipulators. (Text Book1: Chapter 4 and 5)	
UNIT – V	**03 Hours**
DYNAMICS, TRAJECTORY PLANNING AND CONTROL: Dynamic modeling of robots: Lagrange-Euler formulation, Newton-Euler formulation. Trajectory planning: Joint space techniques, Cartesian space formulation. Control of manipulators: PID control, Computed torque control and Force control (Text Book1: Chapter 6,7 and 9,10)	
• Getting Started: Simulating Your First Robot in Isaac Sim • Ingesting Robot Assets and Simulating Your Robot in Isaac Sim • Synthetic Data Generation for Perception Model Training in Isaac Sim • Developing Robots with Software-in-the-Loop (SIL) in Isaac Sim	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy):

Use AI-based simulators (e.g., intelligent robotics environments) to classify robot types and configurations.

- Application of transformations in AI-based computer vision and robot localization.
- AI-assisted kinematic modeling and motion prediction
- AI-based inverse kinematics solvers for real-time robotics Applications.
- Compare human motor control (learning, adaptation) with robotic control systems
- Relate human spatial awareness (navigation, orientation) to coordinate transformations.
- Reflection on human vs robot adaptability in performing tasks
- Reflection on precision vs flexibility in human and robot movements.
- Study of redundancy resolution and optimization in robotic systems
- Reflection on feedback mechanisms in human's vs PID controllers.

AI Tools:

GitHub Copilot, ChatGPT / AI assistants

Applications:

1. Industrial robotic arms in manufacturing and assembly lines. Medical robots for surgery and rehabilitation systems.
2. Robot navigation in autonomous vehicles and drones. Frame transformations in robotic welding and pick-and-place operations
3. Robotic arms in CNC machines and automated assembly systems. Space robotics (e.g., satellite arm manipulation).
4. Robotic path planning in industrial automation. Motion control in humanoid robots and prosthetic devices.
5. Autonomous robots in manufacturing and logistics. Robotic control in drones, self-driving vehicles, and surgical robots.

Mini Project Component

- Some sample projects described below simulate the real software engineering problems.

Project 1 – Robotic Arm Simulation (2–3 DOF)

Design and simulate a robotic arm using Python/MATLAB or

ROS. Key Features:

- Implement forward kinematics
- Visualize arm movement
    - Optional: Add inverse kinematics for position control

Outcome: Hands-on experience in robotic modeling and

simulation. Project 2 – Trajectory Planning and Control

Develop a trajectory planning system for a robotic manipulator.

- Generate joint space trajectory
- Implement basic PID control
- Plot position, velocity, and error graphs

Outcome: Understanding of motion planning and control strategies.

Project 3 – Intelligent Robot Motion Planning

Use AI techniques to optimize robot motion or solve inverse kinematics.

- Apply optimization or simple ML model
- Compare traditional vs AI-based solutions
    - Validate results using simulation

Outcome: Exposure to AI integration in robotics.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain robot structure and configurations	L3
CO2	Apply coordinate transformations in robotics	L3
CO3	Model robotic manipulators using kinematics	L3
CO4	Solve inverse kinematics and velocity problems	L3
CO5	Analyze robot dynamics and implement basic control techniques	L4

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	2	2	–	–	–	–	–	–	3	2
CO2	3	3	2	2	2	–	–	–	–	–	–	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	2	3	2	–	–	–	–	–	–	3	2
CO5	3	3	3	3	3	–	–	–	–	–	–	3	2

3: Substantial (High)   2: Moderate (Medium)   1: Poor (Low)

Textbooks:

1. Introduction to Robotics: Mechanics and Control– John J. Craig, 4th ed., Pearson Education, 2018.Robotics: Modelling, Planning and Control – Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo,1st ed., Springer, 2009.

REFERENCE BOOKS:

5. Robot Modeling and Control– Mark W. Spong, Seth Hutchinson, M. Vidyasagar 1st ed., Wiley, 2006.
6. Introduction to Autonomous Robots– Nikolaus Correll, Bradley Hayes, et al.1st ed., MIT Press, 2022.

E Resources:

1. MIT OpenCourseWare - https://ocw.mit.edu/courses/2-12-introduction-to-robotics-fall-2005/
2. IEEE research articles

Activity Based Learning (Suggested Activities in Class)

- Line Following Robot
- Obstacle Avoidance Robot
- Pick-and-Place Robotic Arm
- Maze Solving Robot
- Human-Following Robot

****

Accelerated Computing with CUDA

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs (15 hrs L + 15 hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision:

The course aims to develop strong foundations in GPU computing and parallel programming using CUDA for designing high-performance computing systems capable of accelerating complex computational workloads. It enables students to understand GPU architecture, CUDA programming model, kernel execution, and memory management techniques for efficient parallel computation. The course also equips learners with practical skills in optimizing CUDA applications and building real-world GPU-accelerated solutions for domains such as artificial intelligence, data science, scientific computing, and real-time processing systems.

Prerequisites: C Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamentals of GPU architecture and parallel computing concepts
2. Develop CUDA programs using kernels and thread hierarchy
3. Apply efficient memory management techniques in CUDA
4. Analyze and optimize CUDA programs for improved performance
5. Design and implement real-world applications using GPU computing.

Broad Course Description

The course Accelerated Computing with CUDA provides a comprehensive understanding of GPU computing and parallel programming using CUDA. It introduces the fundamentals of GPU architecture, parallel computing concepts, and the CUDA programming model, followed by in-depth coverage of kernel execution, thread hierarchy, and execution configuration. The course explores memory management techniques and optimization strategies such as synchronization, atomic operations, and performance tuning. It also includes hands-on development of GPU-enabled applications using CUDA C/C++. Overall, the course equips students with the skills to design, implement, and optimize high-performance computing applications for AI, data science, and real-time systems.

UNIT – I	06 Hours

Introduction to CUDA and GPU Computing:

Why CUDA, GPU vs CPU, Parallel Computing Concepts, CUDA Architecture, Programming Model (Text Book-1: Chapter 1, 2. Text Book: Chapter 1, 2)

NVIDIA DLI courses (Relevant to this module):

1. **An Easier Introduction to CUDA** (1 hr) – CUDA C++, nvcc, nvprof
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-AC-01+V1

UNIT – II	06 Hours

CUDA Kernels and Execution Model:

CUDA kernels and function execution, Thread hierarchy: grid, block, thread, Thread indexing and mapping, Execution configuration
(Text Book-1: Chapter 3. Text Book-2: Chapter 3)

NVIDIA DLI courses (Relevant to this module):

1. Fundamentals of Accelerated Computing with CUDA C++ (8 hrs) – write and launch custom CUDA kernels, control thread hierarchy,

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-AC-04+V2	
UNIT – III	**06 Hours**
Memory Management in CUDA: CUDA memory model, cudaMalloc, cudaMemcpy, Types of memory: global, shared, local, Memory optimization techniques (Text Book 1: Chapter 5. Text Book 2: Chapter 4)	
NVIDIA DLI courses (Relevant to this module): **1.** Fundamentals of Accelerated Computing with CUDA Python (8 hrs)- Multidimensional Grids, and Shared Memory for CUDA Python with Numba https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-AC-10+V1	
UNIT – IV	**06 Hours**
Optimization Techniques: Synchronization and barriers, Atomic operations, Parallel reduction techniques, Performance tuning and profiling (Text Book1: Chapter 6, 7. Text Book2: Chapter 5)	
UNIT – V	**06 Hours**
CUDA Applications and Case Studies: Image processing using CUDA, scientific simulations, CPU vs GPU performance comparison, Case studies and real-world applications (Text Book1: Chapter 8. Text Book2: Chapter 6)	

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **Accelerated AI Computing:** Leverage CUDA-enabled GPUs to accelerate AI and deep learning workloads, enabling faster training and real-time inference for large-scale models.
- **Parallel Intelligence Design:** Utilize thread hierarchies (grid, block, thread) to design scalable AI algorithms that efficiently process high-dimensional data in parallel.
- **Efficient Data Handling & Sustainability:** Optimize memory usage (global, shared, local) to reduce computational overhead and energy consumption, supporting sustainable AI computing practices.
- **High-Performance Model Optimization:** Apply synchronization, atomic operations, and reduction techniques to improve performance of AI algorithms such as matrix operations and neural network training.
- **Real-Time AI Systems:** Use GPU acceleration to build real-time systems for applications like image processing, video analytics, and autonomous systems.
- **Responsible & Explainable AI Systems:** Ensure transparency and fairness by optimizing computational workflows, avoiding bias amplification due to inefficient processing, and enabling reproducible AI experiments.

Suggested Tools:

1. NVIDEA Nsight
2. CUDA Toolkit
3. Numba
4. Google Colab

Applications:

1. Deep learning model training using GPU acceleration
2. Image and video processing (object detection, segmentation)
3. Natural Language Processing (NLP) acceleration
4. Scientific simulations and high-performance computing (HPC)

5. Autonomous systems and real-time analytics
6. Financial modeling and risk analysis
7. Big data processing and parallel data analytics

Mini Project Component

Industry-Driven Projects (NVIDIA / AI Industry Style)
These projects simulate real-world **GPU computing and AI acceleration problems**.

Project 1 – GPU-Accelerated Matrix Operations

1. **Problem**: Implement and optimize matrix multiplication using CUDA
2. **Concepts used**: Kernels, memory optimization, parallel execution
3. **Industry relevance**: Deep learning computations, scientific computing

Project 2 – Image Processing using CUDA

4. **Problem:** Develop a GPU-based image filtering or edge detection system
5. **Concepts used**: Parallel threads, shared memory, optimization
6. **Industry relevance:** Computer vision, healthcare imaging

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain GPU architecture and CUDA programming model	L2
2	Explain GPU architecture and CUDA programming model	L2
3	Develop CUDA programs using kernels and thread hierarchy	L4
4	Apply memory management techniques in CUDA	L3
5	Design CUDA-based solutions for real-world applications	L6

Table: Mapping Levels of COs to POs / PSOs													
COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2			2						2	2	
CO2	3	3	3	2	3				2		2	3	2
CO3	3	3	2	2	3						2	3	2
CO4	2	3	2	3	3						2	3	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Jason Sanders and Edward Kandrot – CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley, 2010.
2. David B. Kirk and Wen-mei W. Hwu – Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 2016.

REFERENCE BOOKS:

- Shane Cook – CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann, 2013
- Toby Webber, "Mastering CUDA C++ Programming: From Fundamentals to Advanced GPU Computing", 2025
- EvoEngineer: Mastering Automated CUDA Kernel Code Evolution with Large Language Models" (arXiv, Oct 2025

E-Resources:

- https://developer.nvidia.com/cuda-zone
- https://docs.nvidia.com/cuda/
- https://nptel.ac.in
- https://colab.research.google.com
- https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-AC-01+V2

Activity Based Learning (Suggested Activities in Class)

Activity 1: Visualization of GPU vs CPU architecture using diagrams and simulations
Activity 2: Hands-on execution of simple CUDA programs (Hello CUDA, vector addition)
Activity 3: Implementation of thread hierarchy and indexing examples
Activity 4: Memory management exercises using cudaMalloc and cudaMemcpy
Activity 5: Case study discussion on real-world CUDA applications (AI, image processing)

****

RESPONSIBLE AI, ETHICS & POLICY

[As per Choice Based Credit System (CBCS)scheme]
SEMESTER - V

Course Code	:		**Credits**	:	02
Hours / Week	:	02	**Total Hours**	:	30 Hours L
L–T–P–J	:	2–0–0–0			

Course Vision:
To equip Undergraduate students with the knowledge and ethical perspective required to design, develop, and deploy trustworthy, fair, and accountable AI systems, while understanding the societal, legal, and policy implications of artificial intelligence.

Course Objectives:
This course will enable students to:
1. Understand ethical frameworks and principles relevant to the design and deployment of artificial intelligence technologies.
2. Analyze the societal, legal, and policy implications of AI adoption across different sectors.
3. Apply the principles of Responsible AI in the development and evaluation of AI systems.
4. Evaluate and design AI solutions that are ethical, transparent, and socially responsible.
5. Design responsible AI strategies and governance mechanisms aligned with global standards and regulatory frameworks.

Broad Course Description:
This course provides a comprehensive understanding of Responsible AI, focusing on the ethical, legal, societal, and governance aspects of artificial intelligence systems. It explores key principles such as fairness, transparency, accountability, and trustworthiness in AI design and deployment across diverse application domains. Students will examine real-world case studies to understand bias, discrimination, privacy concerns, and regulatory challenges associated with AI technologies. The course also introduces global AI governance frameworks, policy guidelines, and risk management strategies to ensure safe and compliant AI adoption. Through conceptual learning and applied analysis, students will develop the ability to critically evaluate AI systems and design responsible, human-centered AI solutions aligned with societal values and international standards.

Teaching-Learning Process (General Instructions)
These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various **Course Outcome**s:
1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the **Course Outcome**s.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Encourage **Collaborative** (Group Learning) Learning in the class.
4. To make **Critical thinking**, asking higher order thinking questions in the class in the form of Quiz and writing programs with complex solutions.
5. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.

Unit– I	06 Hours

Foundation of Responsible and Authentic Intelligence::
Topics: Concept of Responsible AI and **Authentic Intelligence::**, Principles of **Authentic Intelligence::**, Human-centred AI, trust, transparency, fairness, accountability, and inclusivity and

its societal impact, Ethical frameworks for AI, Human values and cultural contexts in AI systems. **Applications:** Ethical decision-making in AI-powered recommendation systems, Responsible deployment of AI assistants and chatbots.	
Unit – II	**06 Hours**
Bias, Fairness, Transparency and Human-Aligned AI **Topics:** Sources of bias in AI, data bias, algorithmic bias, societal bias, Fairness metrics and bias mitigation techniques, Algorithmic transparency and explainability, Explainable AI (XAI) models and interpretability methods. **Applications:** Transparent AI decision systems in government services, Bias detection in hiring and recruitment AI systems.	
Unit– III	**06 Hours**
Privacy, Safety and Trustworthy AI Systems **Topics:** Privacy challenges in AI-driven systems, Privacy-preserving machine learning, differential privacy, federated learning, AI security risks and adversarial attacks, Robustness, safety, and reliability of AI models, Trustworthy AI lifecycle frameworks. **Applications:** Privacy-preserving healthcare analytics, Secure AI-based financial fraud detection systems, autonomous vehicles.	
Unit– IV	**06 Hours**
AI Governance, Regulation and Responsible Innovation **Topics:** AI governance and regulatory frameworks, Algorithmic accountability and auditing mechanisms, AI risk assessment and impact evaluation models, Global AI policy initiatives and standards, Responsible innovation in AI development. **Applications:** AI compliance frameworks in organizations, Ethical review boards for AI systems.	
Unit– V	**06 Hours**
Designing Authentic and Responsible AI Systems **Topics:** Human-centered AI design methodologies, Responsible AI development lifecycle, Social and economic implications of emerging AI technologies, Sustainable and inclusive AI development. **Applications:** Responsible design of generative AI systems, AI solutions for social good, AI deployment in smart cities and governance.	

AI Integration:

1. AI development lifecycle and ethical checkpoints, Ethical considerations in data collection and model design.
2. Fairness evaluation in machine learning pipelines, explainable AI tools for model interpretation, datasets that ensure diversity and representation.
3. Privacy-preserving techniques in ML models, security testing and robustness checks in AI systems, reliability and safe AI systems.
4. Governance mechanisms in AI development processes, Compliance with regulatory standards in AI products, AI auditing tools.
5. Responsible AI frameworks into AI system design, AI models with transparency and accountability mechanisms, monitoring and evaluation of deployed AI systems.

Authentic Intelligence:

1. Human-centered AI design principles, Aligning AI behavior with human values.
2. Value-aligned AI systems, Ethical dataset design and inclusive AI development.
3. Ethical protection of user privacy and digital rights, trust between humans and AI systems.
4. Ethical leadership in AI development, Accountability and transparency in AI-driven decision systems, AI innovation with societal values and public interest.
5. AI systems that respect human dignity and autonomy, fairness, inclusivity, and sustainability in AI innovation, AI systems remain aligned with human values and societal needs.

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain key concepts and principles of Responsible AI, including fairness, transparency, accountability, and privacy	L2
2	Analyze ethical challenges and societal impacts associated with AI technologies	L4
3	Evaluate AI systems for bias, reliability, and ethical risks using appropriate frameworks	L5
4	Interpret AI governance models, regulatory guidelines, and policy frameworks related to responsible AI for ethical governance	L3
5	Apply responsible AI principles in the design and development of trustworthy AI solutions	L4

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs													
COS	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO-1	3	2	1	-	-	3	1	3	-	-	-	2	1
CO-2	2	3	2	2	-	3	2	3	-	-	-	2	1
CO-3	3	3	2	2	3	3	1	3	1	-	-	2	1
CO-	2	3	2	2	2	3	2	3	1	-	-	2	2
CO-5	3	2	3	3	3	3	2	3	2	-	2	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. M. Coeckelbergh - AI ethics, 2020, MIT press, ISBN: 978-0262538190.
2. S. Agarwal & S. Mishra - Responsible AI Implementing Ethical and Unbiased Algorithms, 2021, Springer, https://doi.org/10.1007/978-3-030-76860-7.

REFERENCE BOOKS:

1. S. Voeneky, P. Kellmeyer, O. Mueller, and W. Burgard - The Cambridge handbook of responsible artificial intelligence: interdisciplinary perspectives, eds., 2022, Cambridge University Press, ISBN: 9781009207867 / 9781009207898, DOI: 10.1017/9781009207898.

E-RESOURCES:

1. Virginia Dignum – *Responsible AI.*
https://people.cs.umu.se/virginia/docs/responsibleAI2018.pdfLuciano Floridi – *The Ethics of AI*.
https://www.researchgate.net/publication/385157886_The_Ethics_of_Artificial_In telligence

RECOMMENDED TOOLS

- Programming: Python (basic ML models)
- AI Libraries: Scikit-learn, TensorFlow (conceptual exposure)
- Explainability Tools: SHAP, LIME
- AI Tools: ChatGPT / AI assistants

****

Deep Learning

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	60 Hrs (45Hrs L+15 Hrs P)
L–T–P-J	:	3–0–2-0			

Course Vision: This course develops the ability to design, train, and deploy deep learning models using modern AI frameworks, integrating AI Factory workflows, NVIDIA DLI practices, and ethical AI principles for real-world intelligent systems.

Prerequisites: Linear Algebra, Calculus, Probability & Statistics, Python Programming, Machine Learning, Data handling and Preprocessing

Course Learning Objectives:

This Course will enable students to:

1. Describe the fundamental concepts of artificial neural networks, including perceptrons, multilayer perceptrons, activation functions, and evaluation techniques used in deep learning.
2. Illustrate the learning mechanisms of neural networks, including forward and backward propagation, gradient descent variants, and optimization techniques used in training models.
3. Apply regularization techniques, weight initialization methods, and hyperparameter tuning strategies for building well-generalized deep learning models.
4. Analyze major deep learning architectures such as CNNs, RNNs, LSTMs, GRUs, and attention-based models for vision and sequential data applications.
5. Examine advanced deep learning approaches including autoencoders, graph neural networks (GNNs), and explainable AI techniques with awareness of ethical and responsible AI principles.

Broad Course Description

Deep Learning is a core discipline of artificial intelligence that enables machines to learn hierarchical representations from data. This course covers neural network architectures, optimization techniques, and advanced models such as CNNs, RNNs, with emphasis on model design, training, and evaluation for real-world applications.

Students gain hands-on experience in GPU-accelerated training and high-performance computing workflows inspired by the **NVIDIA Deep Learning Institute.** The curriculum is aligned with industry-recognized certifications, enabling learners to pursue **NVIDIA-Certified Associate credentials in Generative AI (LLMs/Multimodal).** By the end of the course, students will be able to build, optimize, and deploy deep learning models while considering scalability, performance efficiency, ethical implications, and societal impact.

UNIT – I	08 Hours

Introduction to Neural Networks and Deep Learning:

Introduction to Artificial Intelligence, Machine Learning, and Deep Learning: Differences and Relationships.

Foundations of Neural Networks and Deep Learning: Neural Networks, Activation functions, Loss Functions, Hyperparameter, The Perceptron, Multilayer perceptron (MLP).

Engineering Applications: Image classification and object detection, speech recognition, predictive maintenance and fault diagnosis in industrial systems, medical diagnosis and healthcare analytics, autonomous systems and robotics.

UNIT – II	10 Hours
Mathematical Background, Data Preparation and Training Techniques : Data representation using tensors: Scalars, vectors, matrices, tensor shapes, data batches, image data, and introduction to sequence/time-series data representation. The gears of neural networks: Tensor operations—Element-wise operations, Broadcasting, Tensor product, and Tensor reshaping. Training the Model:Forward and Backward Propagation:forward propagation; backpropagation using the chain rule;Training Loop- epochs and batches, forward pass, loss computation, backpropagation, parameter update and validation steps;Gradient descent methods: batch, stochastic, and mini-batch gradient descent;Optimization algorithms: Momentum and Adam. Model Optimization and Generalization: Weight initialization: overview of Xavier/Glorot and He initialization; Regularization techniques: L2 weight decay, dropout, early stopping based on validation performance, and basic concepts of batch normalization; Data augmentation; Hyperparameter tuning using validation data. **Engineering Applications:** Time-series forecasting and demand prediction, anomaly detection in cybersecurity and network monitoring, energy load forecasting in smart grids, traffic flow prediction and intelligent transportation systems, signal processing and sensor data analysis.	
UNIT – III	09 Hours
Convolutional Neural Networks (CNNs) Biological inspiration of CNNs,Convolution operation: kernels/filters, feature maps, stride, and padding (with simple numerical examples). Convolutional layers: parameter sharing, local connectivity, and multiple filters/channels. Pooling layers: max pooling and average pooling; role of batch normalization and dropout in CNNs. CNN architecture: overall structure and working of convolutional neural networks, including input, convolutional, pooling, and fully connected layers. Popular CNN architectures (conceptual overview): LeNet-5 (architecture and idea of early CNN); AlexNet (representation learning—brief—and architecture); VGG (VGG blocks and network design); ResNet (residual blocks and model architecture). Transfer learning: use of pre-trained models (VGG16, ResNet50) for feature extraction and fine-tuning. Object detection (introduction): bounding boxes and YOLO (concept and intuition). **Engineering Applications:** Autonomous driving and ADAS, remote sensing and satellite imagery, biometric authentication (fingerprint/iris recognition), agricultural monitoring using drones, and CNN-based access control systems (e.g., NVIDIA DLI case study: “Personalized Doggy Door”).	
UNIT – IV	09 Hours
Recurrent Neural Networks, LSTMs and Attention Mechanism : Need for Sequential Models: limitations of MLP and CNN for sequential data (text, time-series, audio). Recurrent Neural Networks (RNNs): architecture, hidden state, and unrolling through time. Vanishing Gradient Problem: challenges in learning long-term dependencies in RNNs. Long Short-Term Memory (LSTM): cell state, input, forget, and output gates. Gated Recurrent Unit (GRU): simplified gating mechanism and comparison with LSTM. Attention Mechanism: limitations of fixed context vectors in Seq2Seq models; queries, keys, and values; Bahdanau (additive) attention-conceptual understanding. **Engineering Applications:** Machine translation systems, predictive maintenance using sensor sequence data, anomaly detection in financial transactions, human activity recognition using wearable sensors, network traffic analysis and intrusion detection.	
UNIT – V	09 Hours
Unsupervised Deep Learning and Applications :	

Unsupervised Pretrained Networks (UPNs): motivation and learning from unlabelled data.
Autoencoders: encoder–decoder architecture, bottleneck/latent space, applications in compression and denoising, Variational Autoencoders, Introduction to Generative Adversarial Networks (GANs)
Graph Neural Networks (GNNs): basic concepts of graph-structured data, learning on graphs, and applications.
Explainable AI (XAI): importance of model interpretability; Grad-CAM, LIME, and SHAP concepts.
Responsible AI and Ethics: bias in datasets, fairness, data privacy, and societal impact of AI.

Engineering Applications: Graph-based analysis for smart city infrastructure and networked systems, data compression and denoising in communication and signal processing, explainable AI for reliable decision-making in critical systems, and development of fair, unbiased, and privacy-preserving AI solutions.

NVIDIA DLI Course Mapping

1. Getting Started with Deep Learning
2. Fundamentals of Deep Learning(workshop)
3. Introduction to Graph Neural Networks.

Nvidia Learning Path

1. Deep Learning
2. Gen AI LLM/Generative AI Multimodal)

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration:**
  - **Module 1:** AI-assisted model development and visualization tools.
  - **Module 2:** AI-based hyperparameter tuning, AutoML tools, and AI-assisted performance analysis.
  - **Module 3:** AI-based vision tools, use of pre-trained models, and transfer learning with foundation models.
  - **Module 4:** NLP frameworks (Transformers), introductory LLM APIs, and AI-assisted text generation tools.
  - **Module 5:** AI development and deployment tools, AI factory platforms, and MLOps automation tools.
- **Authentic Intelligence Reflection:**
  - **Module 1:** limitations of neural networks and interpretability challenges.
  - **Module 2:** bias in training data, overfitting risks, and reliability and fairness concerns.

- **Module 3:** ethical issues in facial recognition, dataset bias in vision systems, and risks in safety-critical automated decisions.
- **Module 4:** misuse of generative AI, hallucination and misinformation risks, and ethical concerns in NLP systems.
- **Module 5:** responsible AI deployment, governance and accountability, and safety, transparency, and trustworthy AI systems.

Laboratory Component

Experiments will be conducted using **PyTorch** or **TensorFlow** on **Jupyter Notebook** or **Google Colab**, following workflows inspired by the **NVIDIA Deep Learning Institute (DLI)**. Students will implement models, evaluate performance using metrics such as accuracy and loss, and analyze training behavior. NVIDIA ecosystem tools such as transfer learning toolkits for vision tasks (**e.g., NVIDIA TAO Toolkit**) and language/speech model development frameworks (**e.g., NVIDIA NeMo**) may be used.

Hardware Requirement: Internet-enabled system with access to a GPU-accelerated computing environment (NVIDIA GPU cloud or equivalent).

List of Experiments

1. Create tensors and perform basic operations such as reshaping, indexing, and element-wise computations using a deep learning framework. Demonstrate simple forward computation without training.
2. Implement linear and logistic regression models using gradient descent. Build a basic training loop with forward pass, loss computation, and parameter updates. Evaluate model performance using appropriate metrics.
3. **Training a Basic Neural Network**

   Train a neural network for image classification using the MNIST dataset and evaluate performance using loss and accuracy. Implement the model using NumPy/PyTorch and analyze training behavior using loss curves.
4. **Convolutional Neural Network (CNN)**

   Design and train a Convolutional Neural Network (CNN) for image classification using MNIST or American Sign Language dataset. Compare performance in terms of accuracy and feature learning.
5. **Data Augmentation**

   Apply data augmentation techniques such as rotation, flipping, and scaling to enhance the dataset and evaluate their impact on model generalization and performance.
6. **Optimization and Regularization**

   Train a neural network using different optimizers (SGD, Adam) and apply regularization techniques such as dropout and batch normalization. Compare overfitting using training and validation curves.
7. **Using Pre-trained Models**

Implement an image classification system using a pre-trained model (e.g., ResNet). Perform inference on new images and understand how classification layers work in deep learning models.

8. **Transfer Learning**

   Fine-tune a pre-trained model on a custom dataset and compare performance with training from scratch. Analyze convergence speed and generalization.

9. **Sequence Modeling using RNN/LSTM**

   Train an RNN or LSTM model for sequence prediction (text or time-series). Evaluate performance and understand handling of sequential data.

10. **Natural Language Processing using Pre-trained Models**

    Implement an NLP task such as text classification using a pre-trained language model. Perform tokenization and text segmentation during preprocessing and understand text masking in masked language modeling. Evaluate model performance using accuracy and loss. *(e.g., sequence modeling / attention-based models, machine translation systems, BERT-based NLP models)*

11. **GNN**

    Implement a Graph Neural Network (GNN) for node classification on a graph dataset. Represent data using nodes and edges, train the model efficiently, and evaluate performance. Analyze the impact of graph structure and discuss real-world applications.

Project Component *(Indicative, not limited to)*

1. Develop an image classification model that interprets color images using deep learning techniques (CNN-based approach).
2. Design and implement a data generator with augmentation techniques to effectively train models on small datasets and improve generalization.
3. Build an efficient image classification system using transfer learning and feature extraction with pre-trained models (e.g., ResNet), and analyze improvements in training speed and accuracy.
4. Implement a deep learning solution that combines data augmentation and transfer learning to optimize model performance on limited data scenarios.
5. Explore and compare advanced neural network architectures (e.g., CNN variants, transformers) for image-based tasks and analyze their performance.
6. Design an end-to-end deep learning pipeline for image-based applications, including data preparation, model training, and performance evaluation.
7. Investigate recent advancements in deep learning architectures and apply one technique to improve model performance on a selected task.

Research Component:

Students will select three recent research papers in the deep learning domain from reputed sources such as IEEE, Springer, or arXiv, and conduct a literature survey by analyzing the problem, methodology, and key findings. A brief report and presentation summarizing insights and comparisons will be submitted.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	**Outline** fundamental concepts of deep learning and neural networks.	L2
CO2	**Apply** forward and backward propagation, tensor operations, and optimization techniques for training deep learning models.	L3
CO3	**Construct and develop** deep learning models using regularization, weight initialization, and hyperparameter tuning to improve generalization.	L3
CO4	**Analyze** and compare architectures such as CNNs, RNNs, LSTM, GRU, and attention-based models for solving vision and sequential data problems.	L4
CO5	**Examine** deep learning solutions using autoencoders, GNNs, and explainable AI techniques while considering ethical and responsible AI principles.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	1	1	2	–	–	1	2	1	2	3	2
CO2	3	3	2	1	3	-	–	1	2	2	3	3	2
CO3	2	3	3	2	3	-	1	2	2	2	3	3	2
CO4	2	3	2	2	3	2	-	2	2	2	3	3	2
CO5	1	2	2	2	2	3	3	2	2	2	3	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Assessment Scheme: Integrated Professional Core Courses (Theory + Practical)

Criteria	**Weightage of Marks**
Continuous Assessment Tests (CAT) – includes mini-project/research component	10
industry-oriented certification (NVIDIA DLI or equivalent)	10
Mid Semester Examination (MSE)	20
Lab Continuous Evaluation and Lab Record	10
Lab exam	10
Total	**60 marks**

Suggested Reading

Primary Textbooks

1. Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola, "Dive into Deep Learning", Cambridge University Press,ISBN-10 **:** 1009389432,2023**.**
2. Josh Patterson and Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Media, ISBN:978-1-4919-1425-0, 2017.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning (Adaptive Computation and Machine Learning series", MIT Press, 2017.

Additional Resources

1. Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow", O'Reilly, 3rd Edition, 2022.
2. Francois Chollet, "Deep Learning with Python", Manning Publications, 2nd Edition, 2021.
3. Simon Haykin, "Neural Networks: A Comprehensive Foundation", 2nd Edition, Prentice Hall, 2003.
4. Nikhil Buduma, "Fundamentals of Deep Learning", O'Reilly Media, 2017.

E-Resources

1. NPTEL Deep Learning: https://onlinecourses.nptel.ac.in/noc20_cs62/preview
2. Dive into Deep Learning (interactive): https://d2l.ai

NVIDIA Deep Learning Institute: https://learn.nvidia.com/

****

Computer Networks & Distributed Systems

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	30 HrsL+15HrsT+15 HrsP)
L–T–P-J	:	2–1–2-0			

Course Vision: The course aims to develop strong foundations in computer networks and distributed systems with a focus on their application in modern AI infrastructure and scalable cloud-based services. It enables students to understand networking models, protocols, routing mechanisms, and communication technologies for designing low-latency, high-performance AI systems and APIs. The course also equips learners with practical knowledge of distributed computing principles, streaming architectures, and GPU communication frameworks required to optimize AI model deployment, inference, and global-scale system performance.

Prerequisites: Programming in any Language, Networking Basics

Course Learning Objectives:

This Course will enable students to:

1. Understand and compare OSI and TCP/IP models, analyzing their relevance to modern AI API latency and streaming.
2. Apply IP addressing, subnetting, and routing protocols (OSPF, BGP) to design scalable, low-latency cloud AI topologies.
3. Evaluate transport and application protocols (TCP, UDP, HTTP/2, WebSockets) for reliability, efficiency, and secure AI communication.
4. Implement streaming LLM response endpoints using Server-Sent Events (SSE) in Python.
5. Analyze DNS, CDN, and load balancing strategies to optimize global AI API distribution and cache performance.
6. Apply CAP theorem and consensus algorithms (Raft) to distributed ML systems, balancing availability, consistency, and reliability.
7. Benchmark gRPC vs REST communication models for high-performance AI inference APIs under heavy load.
8. Examine GPU communication frameworks (NVIDIA NCCL, NVLink) to understand high-throughput distributed training.

Broad Course Description

This course explores how classical network models and protocols underpin modern AI infrastructure, focusing on latency, scalability, and distributed communication. Students will study OSI and TCP/IP layers, IP addressing, routing protocols, and transport/application mechanisms like TCP, UDP, HTTP/2, and WebSockets, applying them to real-world AI API design and streaming LLM responses. The curriculum extends to DNS, CDN, and load balancing for global distribution, while also examining distributed systems trade-offs through the CAP theorem and consensus algorithms. Finally, learners will compare gRPC and REST for high-performance AI services and analyze GPU communication frameworks such as NVIDIA NCCL and NVLink, connecting textbook theory to practical cloud AI deployment and optimization.

.

UNIT – I	05 Hours

Network Models & Latency: Network Fundamentals & Protocols

OSI and TCP/IP models — connect to AI API network latency analysis; why HTTP/2 multiplexing matters for LLM streaming

Textbook 1: Ch. 1 (Introduction), Ch. 2 (Physical Layer), Ch. 6 (Transport Layer) Textbook 2: Ch. 2 (Data Transmission), Ch. 3 (Data Communication Protocols)

UNIT – II	05 Hours
Addressing & Routing: IP addressing, routing (OSPF, BGP), subnetting — design the network topology for a cloud AI startup (multi-region, low-latency) Textbook 1: Ch. 5 (Network Layer: Routing) Textbook 2: Ch. 5 (Network Layer: Addressing & Routing)	
UNIT – III	**05 Hours**
Transport & Application Protocols: TCP, UDP, HTTP, HTTPS, WebSockets — implement a streaming LLM response endpoint with SSE (Server-Sent Events) in Python Textbook 1: Ch. 6 (Transport Layer), Ch. 7 (Application Layer) Textbook 2: Ch. 6 (Transport Layer Protocols), Ch. 7 (Application layer)	
UNIT – IV	**07 Hours**
Application Services & Distribution: DNS, CDN, load balancing — design a globally distributed AI API with Cloudflare CDN; analyse cache hit rates Textbook 1: Ch. 7 (Application Layer: DNS, Web, CDN) Textbook 2: Ch. 7 (Application Layer Services)	
UNIT – V	**08 Hours**
Distributed Trade-offs Distributed Systems for AI CAP theorem — understand trade-offs in distributed vector databases (availability vs consistency) NVIDIA NCCL: distributed GPU communication — understand how DGX B200 NVLink 4 enables 900 GB/s all-reduce for distributed training API Communication Models and High Performance GPU Communication gRPC vs REST for AI services — benchmark gRPC vs REST for LLM inference API at 1000 req/s Textbook 1: Ch. 7 Distributed consensus: Raft algorithm — how distributed ML parameter servers stay consistent Textbook 2: Ch. 8 (Network Security & Reliability touches distributed coordination) Textbook 1: Ch. 7 (Application Layer: RPC concepts) Textbook 2: Ch. 7 (Application Layer Services)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Globally distributed AI API service
- Build: (1) LLM inference endpoint with async streaming (FastAPI + SSE), (2) load balancer (nginx) distributing to 2 model replicas, (3) Redis caching for repeated queries, (4) Prometheus + Grafana monitoring dashboard. Benchmark throughput, latency, cache efficiency. Frame as "infrastructure design for a funded AI startup".

Applications:

- **Latency Models**: Benchmark AI API performance and optimize streaming with modern protocols like HTTP/2.
- **Addressing & Routing**: Guide the design of multi-region cloud topologies to achieve low-latency inference.
- **Transport & Application Protocols**: Enable real-time LLM endpoints using Server-Sent Events (SSE) or WebSockets.
- **DNS, CDN & Load Balancing**: Support globally distributed AI APIs with efficient caching and traffic management.
- **Distributed Systems Trade-offs**: Concepts like the CAP theorem and Raft consensus underpin vector databases and parameter server consistency.

- **API Communication Models**: Compare gRPC versus REST for efficient communication.
- **GPU Communication Frameworks**: Tools like NVIDIA NCCL drive high-performance inference and training pipelines.

Lab Component

1. **Latency & Congestion Simulation** Implement a three-node point-to-point network with duplex links. Vary bandwidth and queue size to measure packet drops and analyze latency trade-offs.
2. **Routing & Path Optimization** Write a program to compute shortest paths between nodes using the Bellman-Ford algorithm. Extend the experiment to simulate routing in a multi-region AI service topology.
3. **Transport Protocols & Streaming** Using TCP/UDP sockets, build a client-server program where the client requests a file and the server streams its contents. Compare performance with datagram sockets for real-time message delivery.
4. **Wireless & Mobile Systems** Simulate a wireless LAN (ESS) with transmitting nodes and evaluate packet transmission performance. Extend to study GSM/CDMA behavior using NS2/NS3 or equivalent environments.
5. **Reliability & Security in Communication** Implement error detection using CRC codes and a simple RSA algorithm for encryption/decryption. Demonstrate how these ensure integrity and confidentiality in distributed AI APIs.
6. **Congestion Control & Distribution** Implement congestion control using the leaky bucket algorithm. Analyze throughput and packet drops, connecting results to CDN/load balancing strategies in globally distributed systems.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	**Analyze** network fundamentals and protocol behavior using OSI/TCP-IP models, and evaluate latency impacts in AI API communication, including the role of HTTP/2 multiplexing in LLM streaming.	L2
2	**Design** and configure IP addressing and routing schemes (OSPF, BGP, subnetting) to build multi-region, low-latency network topologies suitable for cloud AI startups.	L3
3	**Implement** and compare transport/application protocols (TCP, UDP, HTTP, HTTPS, WebSockets) by building streaming endpoints (e.g., SSE in Python) and assessing their performance in real-time AI services.	L3
4	**Apply** DNS, CDN, and load balancing mechanisms to design globally distributed AI APIs, and analyze cache hit rates for optimizing service delivery.	L4
5	**Evaluate** distributed system trade-offs using the CAP theorem and consensus algorithms (Raft), and assess consistency vs availability in distributed vector databases and ML parameter servers. Benchmark API communication models (gRPC vs REST) under high request loads, and analyze high-performance GPU communication frameworks (e.g., NVIDIA NCCL, NVLink) for scalable distributed training.	L5

COS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	3							2	3	3
CO2	3	3	3	3							2	3	3
CO3	3	3	2	3							2	3	3
CO4	3	3	3	3							3	3	3
CO5	3	3	3	3							3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXTBOOKS

1. Computer Networks - Andrew S Tanenbaum, 5th Edition, Pearson Education.
2. Behrouz A. Forouzan, Data Communications and Networking, Fifth Edition TMH, 2013.

Activity Based Learning (Suggested Activities in Class)

- Globally distributed AI API service
- Build: (1) LLM inference endpoint with async streaming (FastAPI + SSE), (2) load balancer (nginx) distributing to 2 model replicas, (3) Redis caching for repeated queries, (4) Prometheus + Grafana monitoring dashboard. Benchmark throughput, latency, cache efficiency. Frame as "infrastructure design for a funded AI startup".

****

AI-Enabled Software Systems & Agile Project Management

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours L
L–T–P-J	:	3–0–0-0			

Course Vision: The course aims to develop strong foundations in AI-enabled software engineering and Agile development practices for building scalable, intelligent, and production-ready systems. It enables students to design AI-integrated software architectures, apply Agile methodologies for iterative development, and manage end-to-end AI system lifecycles including data processing, deployment, and evaluation. The course also equips learners with practical skills in MLOps, collaborative development tools, and ethical AI practices required to deliver robust and adaptive intelligent software solutions in real-world environments.

Prerequisites: Basics of Programming (Python/Java) , Software Engineering Fundamentals, Introduction to Artificial Intelligence (preferred)

Course Learning Objectives:

This course will enable students to:

1. Understand AI-enabled software architectures and intelligent system design.
2. Apply Agile methodologies for managing AI-driven projects.
3. Develop AI-integrated software solutions using modern tools and frameworks.
4. Analyze lifecycle challenges in AI systems including data, deployment, and ethics.
5. Implement collaborative, iterative, and adaptive development practices

Broad Course Description

This course focuses on the integration of Artificial Intelligence into modern software systems and the application of Agile methodologies for efficient project management. It covers AI-driven system architecture, data-centric development, MLOps pipelines, and real-time deployment strategies. Students will explore Agile frameworks such as Scrum, Kanban, and SAFe, and understand how iterative development aligns with AI experimentation cycles. The course emphasises hands-on learning using industry-relevant tools including cloud AI platforms, Git-based workflows, CI/CD pipelines, and collaboration tools. By bridging AI technologies with Agile practices, the course prepares students to build scalable, ethical, and production-ready intelligent systems in multidisciplinary environments

UNIT – I	09 Hours
Foundations of AI-Enabled Software Systems: AI in Software Engineering: Evolution and impact**,** AI system components: Data, models, pipelines**,** Software architecture for AI systems (microservices, APIs) **,** Data-driven development lifecycle**,** Introduction to MLOps concepts *(Text Book 1: Chapters 1,2 | Ref 2,3)*	
UNIT – II	**09 Hours**
Agile Methodologies & Frameworks: Agile Manifesto and principles**,** Scrum framework: roles, artefacts, ceremonies**,** Kanban and Lean development**,** SAFe for scaling Agile**,** Agile vs Traditional project management *(Text Book 2: Chapters 1–4 | Ref 1)*	

UNIT – III	09 Hours
AI Development Lifecycle & Tools: Data collection, pre-processing, labelling**,** Model development and evaluation**,** Version control for AI (Git, DVC) **,** CI/CD for AI pipelines**,** Experiment tracking and reproducibility *(Text Book 3: Chapters 3,5 | Ref 2,4)*	
UNIT – IV	**09 Hours**
Deployment, Monitoring & Ethics: Model deployment (cloud, edge, APIs), Monitoring AI systems (drift,	

performance), Responsible AI: bias, fairness, transparency, Legal and ethical aspects (link to IPR concepts) Security in AI systems *(Text Book 1: Chapter 6 | Ref 3,5)*	
UNIT – V	**09 Hours**
Agile AI Project Execution: Managing AI projects in Agile sprints ,Product backlog for AI features ,Risk management in AI projects ,Collaboration and interdisciplinary teams ,Case studies (industry: Google, Amazon, Tesla AI systems) *(Text Book 2,3 | Ref 4,5)*	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-assisted coding using GitHub Copilot
- Experimentation using Google Colab
- Model deployment via Amazon Web Services / Microsoft Azure
- MLOps pipelines using MLflow .
- Agile tracking using Jira

Applications

1. AI-powered enterprise systems
2. Intelligent SaaS platforms
3. Healthcare AI solutions
4. FinTech automation systems
5. Autonomous and smart Applications

Mini Project Component

Project 1 — AI Agile Sprint Simulation Build a small AI product using Scrum methodology
- Deliverables: backlog, sprint reports, working prototype

Project 2 — MLOps Pipeline Development

- Develop and deploy an AI model with CI/CD
- Tools: GitHub, MLflow, Docker

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain AI-enabled software architectures and Agile principles	L2
CO2	Apply Agile methodologies to manage AI projects	L3
CO3	Develop AI-integrated software systems using modern tools	L3
CO4	Analyze deployment, monitoring, and ethical challenges in AI	L4
CO5	Design scalable AI systems using Agile practices	L6

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2								2		3	2
CO2	2	3	3		2					3		2	3
CO3	2	2	3	2	3					2		2	2
CO4	2	3	2	3	2	2		2		2		2	3
CO5	3	3	3	2	3	2		2	3	3	2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1.Designing Machine Learning Systems, Chip Huyen, O'Reilly Media, 1st Edition, 2022.
2.Agile Project Management with Scrum, Ken Schwaber, Microsoft Press, 1st Edition, 2004.
3.Machine Learning Engineering, Andriy Burkov, True Positive Inc., 1st Edition, 2020.

REFERENCE BOOKS:

1.The Lean Startup, Eric Ries, Crown Business, 1st Edition, 2011.
2.Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, O'Reilly Media, 2nd Edition, 2019.
3.Site Reliability Engineering, Google, O'Reilly Media, 1st Edition, 2016.
4.Accelerate: The Science of Lean Software and DevOps, Nicole Forsgren, Jez Humble, Gene Kim, IT Revolution Press, 1st Edition, 2018.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Fundamentals of Accelerated Computing with CUDA C/C++	Unit III & IV	Relates to AI model development, parallel computing, and deployment optimization using GPUs.
Fundamentals of Deep Learning for Multi-GPU and CPU Systems	Unit I & V	Connects AI system architecture with scalable training and Agile-based AI project execution.
Edge AI and Embedded Systems with Jetson	Unit IV	Maps to AI model deployment on edge devices and real-time intelligent systems.
Fundamentals of Accelerated Data Science	Unit I & III	Links data-driven development, preprocessing, and AI pipelines with high-performance computing.
Building Real-Time Video AI Applications	Unit IV & V	Demonstrates deployment, monitoring, and Agile delivery of real-time AI Applications.

NVIDIA DLI Course / Content	Mapped Units	Relevance
MLOps: Production Machine Learning Systems	Unit III & V	Covers CI/CD pipelines, experiment tracking, and lifecycle management of AI systems in Agile environments.

E-RESOURCES:

1. NPTEL – AI & Software Engineering courses
2. Coursera – Agile & AI Specialisations
3. edX – MLOps and AI systems
4. Google Cloud AI Learning Paths
5. NVIDIA DLI Courses

Activity-Based Learning

1. Agile sprint simulation (team-based)
2. AI model building hackathons
3. CI/CD pipeline demonstrations
4. Case study discussions (industry AI failures & success)
5. Peer code reviews using Git

****

Small E- Business Launch

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	02
Hours / Week	:	01 Hour	**Total Hours**	:	30 Hrs (15Hrs L+15 Hrs J)
L–T–P-J	:	1–0–0-2			

Course Vision: The course aims to develop entrepreneurial and digital business skills required to identify market opportunities, validate innovative ideas, and launch sustainable e-business ventures in the modern digital economy. It enables students to understand customer needs, business models, feasibility analysis, and digital trade practices for building successful online enterprises. The course also equips learners with practical knowledge in lean start-up methodologies, digital platforms, business planning, and legal and operational aspects necessary for managing technology-driven entrepreneurial ventures effectively.

Course Learning Objectives:

This Course will enable students to:

1. To introduce students to the fundamentals of entrepreneurship and digital business transformation.
2. To develop the ability to identify customer needs and market opportunities.
3. To equip learners with skills for idea generation, validation, and feasibility analysis.
4. To understand business models, strategies, and planning techniques.
5. To familiarize students with legal, operational, and digital trade aspects of entrepreneurship.

Broad Course Description

The course Small E-Business Launch provides a comprehensive understanding of the entrepreneurial process in the digital era, focusing on identifying customer needs, generating and validating business ideas, and developing viable digital business opportunities. It covers essential aspects such as business models, feasibility analysis, and lean start-up methodologies, along with practical skills in wireframing, service-market fit, and logistics. The course also emphasizes legal considerations, digital platforms, and emerging practices like virtual trade shows, enabling students to design, launch, and manage sustainable e-business ventures using modern tools and strategies.

UNIT – I	**03 Hours**
INTRODUCTION: Introduction, Identifying Customer Needs, Opportunity Identification & Planning, Idea Generation and Validation, Entrepreneurial Culture, Digital Business Opportunity.	
UNIT – II	**03 Hours**
Business Plan & its Components, Business Model and Strategies, Selection of Appropriate Business Model, Business Model Canvas, Feasibility.	
UNIT – III	**03 Hours**
Lean thinking, Specifications, Wireframe Generation and Validation.	
UNIT – IV	**03 Hours**
Service-Market Fit, Logistics of Goods Delivery, Virtual Trade Shows ,	
UNIT – IV	**03 Hours**
Introduction to Entrepreneurial Law , Legal Aspects of E-Business	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-driven Opportunity Identification:** Use AI tools for customer need analysis and idea validation, while ensuring ethical data usage and unbiased decision-making.

- **Intelligent Business Modelling:** Apply AI-based analytics for Business Model Canvas and feasibility studies with transparent and explainable assumptions.
- **Lean AI Prototyping:** Integrate AI tools for wireframing, rapid prototyping, and validation, ensuring sustainable and resource-efficient development.
- **AI-enabled Market Fit Analysis:** Use AI for predicting service-market fit and customer behaviour, with a focus on privacy-aware and responsible data handling.
- **Smart Logistics & Digital Platforms:** Implement AI in logistics optimization and virtual trade systems while promoting fairness, security, and sustainability in digital operations.

Suggested Tools:

- Perplexity AI
- Shopify
- Canva Magic Design
- Mailchimp AI
- Google Analytics

Applications:

1. Analyze customer needs and behaviour using AI while ensuring ethical and privacy-aware data usage.
2. AI-powered Business Model Optimization.
3. AI-driven Prototyping & Wireframing Tools.
4. AI-enabled Market Prediction Systems.
5. AI in Smart Logistics & Supply Chain.

Project Component

- **Healthcare & Wellness:** Online consultation, fitness platforms, mental health support, pharmacy delivery.
- **Education & EdTech:** E-learning platforms, skill development portals, tutoring services.
- **Food & Hospitality:** Online food delivery, cloud kitchens, restaurant booking systems.
- **Logistics & Supply Chain:** Delivery services, inventory management, smart logistics solutions.
- **FinTech & Digital Payments:** Budgeting apps, digital wallets, financial advisory platforms.
- **Agriculture & Rural Tech:** Farm-to-market platforms, agri-product sales, farmer support systems.
- **Sustainable & Green Business:** Eco-friendly products, recycling platforms, sustainable services.
- **Service-Based Platforms:** Home services (cleaning, repair), freelancing, booking systems.
- **Social & Community Platforms:** Local business promotion, event

management, community marketplaces.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Identify customer needs, analyze opportunities, and generate validated ideas for digital business Applications.	L4
CO2	Develop a business plan and design appropriate business models using tools like Business Model Canvas and feasibility analysis.	L6
CO3	Apply lean thinking principles to design, prototype, and validate solutions through specifications and wireframes.	L3
CO4	Analyze logistics and implement and present a sustainable e-business solution.	L4
CO5	Evaluate service–market fit and legal aspects required for developing a sustainable e-business solution.	L5

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	3	2	2	3	2		2	2	2	2		3
CO2	3	3	3	2	3	2		2	2	3	3		3
CO3	2	3	3	2	3	2		2	2	2	2		3
CO4	2	3	3	3	3	3	2	2	2	3	3		3
CO5	2	2	2	2	2	2	2	2	2	2	2		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Fisher, Steven, Duane, Ja-Nae (2016). Start-up Equation A Visual Guidebook for Building your Start- up, New Delhi: Tata McGraw-Hill.
2. Ulrich, K. T. (2019). Product design and development. Tata McGraw-Hill Education, 7th edition.

REFERENCE BOOKS:

1. Holt H., David (2017). Entrepreneurship: New Venture Creation, New Delhi: Pearson Education.
2. Zimmerer, W. Thomas and Scarborough, M. Norman and Doug Wilson (2009). Essentials of Entrepreneurship and Small Business Management, 5/e; New Delhi: Prentice Hall India.

E-RESOURCES:

1. https://onlinecourses.nptel.ac.in/noc24_mg92/preview
2. https://alison.com/tag/business-management
3. https://www.coursera.org/specializations/start-your-own-business
4. https://www.coursera.org/projects/facebook-small-business-marketing
5. https://www.udemy.com/course/launching-your-small-business-in-india/

Activity Based Learning (Suggested Activities in Class)

1. Proposal Handout & Presentation
a. The Problem
b. Need
c. Existing Alternatives
d. Potential Market

2. Wireframe Sketches, Target Specifications
a. List Critical Customer Needs
b. For a., Prepare a List of Target Specifications
c. Prepare Wireframe

****

Design Studio III: Digital–Physical Systems

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hrs (15 Hrs L+15 Hrs P)
L–T–P-J	:	1–0–0-2			

Course Vision: This course aims to develop a strong foundation in Cyber-Physical Systems by integrating embedded systems, IoT technologies, and artificial intelligence for real-world applications. It aims to enable students to design and prototype intelligent systems that can sense, communicate, and make data-driven decisions. The course further aims to prepare learners to solve interdisciplinary engineering problems in domains such as smart agriculture, healthcare, and energy systems through practical, innovation-driven CPS development.

Prerequisites: Basic Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. To build foundational knowledge of sensors, actuators, and embedded systems
2. To enable system-level integration of hardware and software components
3. To develop IoT-enabled data acquisition and communication systems
4. To introduce intelligent decision-making using AI/ML techniques
5. To foster innovation through real-world problem-driven CPS design.

Broad Course Description

This course focuses on the design and rapid prototyping of Cyber-Physical Systems (CPS) by combining embedded systems, IoT, and AI. Students learn to acquire real-world data using sensors, process it using microcontrollers, transmit it through IoT protocols, and apply intelligent decision-making using lightweight AI models.

The course culminates projects addressing real-world challenges such as smart agriculture, energy systems, and healthcare monitoring.

UNIT – I: RAPID PROTOTYPING FOUNDATIONS (Weeks 1–3)	
Students learn how cyber-physical systems sense, process, and act on the real world, then wire their first sensors and actuators to an Arduino or ESP32. The unit closes by demystifying signal acquisition and conditioning so noisy analog data becomes reliable digital input.	**Authentic intelligence:** Before accepting any AI-generated wiring diagram or code snippet, ask: does this match the actual datasheet for my sensor, or is it a plausible-looking hallucination? Verify every pin number, voltage level, and resistor value against the manufacturer's datasheet — AI tools frequently confuse similar part numbers and produce confidently wrong circuit suggestions.
Week 1: Introduction to Digital Physical Systems / CPS	
LECTURE (1 HR) Overview of Digital Physical Systems (DPS) and Cyber-Physical Systems (CPS). Components, architecture, and real-world applications in smart cities, healthcare, manufacturing, and	**PROJECT TASK (2 HRS)** Identify and study three real-world CPS applications. Create a system architecture diagram showing sensors, controllers, communication modules, and actuators. Present findings to peers.

transportation.	
Week 2: Sensors and Actuators	
LECTURE (1 HR) Types of sensors and actuators, selection criteria, interfacing requirements, calibration, and performance characteristics.	**PROJECT TASK (2 HRS)** Interface one sensor and one actuator with Arduino/ESP32. Collect sensor readings and control actuator behavior based on predefined thresholds.
Week 3: Microcontroller Basics & Signal Acquisition	
LECTURE (1 HR) Arduino/ESP32 architecture, GPIO fundamentals, analog and digital signals, signal acquisition and conditioning techniques.	**PROJECT TASK (2 HRS)** Develop a sensor-based monitoring system using Arduino/ESP32. Acquire, process, and display sensor data through serial monitoring or visualization tools.
UNIT – II: EMBEDDED INTEGRATION (Weeks 4–6)	
Students master the three core communication primitives — GPIO, ADC, and PWM — and learn to combine multiple sensors on shared I2C and SPI buses without conflict. Power budgeting and a structured debugging workflow ensure the system is both reliable and fixable when things go wrong.	**Authentic intelligence:** AI coding tools generate boilerplate I2C and SPI code quickly, but they often hardcode device addresses without checking for conflicts or omit pull-up resistor requirements. Run every AI-generated protocol snippet through the component datasheet before flashing — a missing 4.7kΩ pull-up or a duplicate address will silently corrupt your bus and produce data that looks real but isn't.
Week 4: GPIO and ADC Interfacing	
LECTURE (1 HR) Lecture 1: Digital and analog interfacing fundamentals using GPIO and ADC modules	**PROJECT TASK (2 HRS)** Design and implement a digital control application using LEDs, switches, and sensors.
Week 5: PWM and Actuator Control	
LECTURE (1 HR) PWM generation, motor control, brightness control, and embedded output applications.	**PROJECT TASK (2 HRS)** Control the speed of a DC motor or LED brightness using PWM signals generated by Arduino/ESP32.
Week 6: Multi-Sensor Integration and Debugging	
LECTURE (1 HR) Multi-sensor systems, power management, reliability considerations, debugging strategies, and troubleshooting.	**PROJECT TASK (2 HRS)** Integrate multiple sensors into a single platform, test functionality, analyze power consumption, and perform debugging.
UNIT – III: CONNECTIVITY AND DATA FLOW (Weeks 7–9)	
Students connect their embedded device to the internet using MQTT and HTTP, then build a data pipeline that logs locally at the edge and visualizes live on the cloud. A dedicated security class ensures students understand where credentials can leak and how to encrypt data in	**Authentic intelligence:** AI assistants routinely produce IoT connection examples with Wi-Fi passwords and API keys written directly in source code — a critical security flaw. Before using any AI-generated connectivity snippet,

transit.	check that credentials are loaded from environment variables or a config file excluded from version control, and that the channel uses TLS. Never commit what an AI gave you without a security review.

Week 7: IoT Communication Protocols

LECTURE (1 HR)	**PROJECT TASK (2 HRS)**
Introduction to MQTT, HTTP, client-server communication, and message exchange.	Configure a device to publish and subscribe to data using MQTT or transmit sensor data using HTTP.

Week 8: Edge and Cloud Computing Concepts

LECTURE (1 HR)	**PROJECT TASK (2 HRS)**
Edge processing, cloud services, distributed systems, advantages and limitations.	Implement local processing and cloud-based data storage, then compare performance metrics.

Week 9: Data Logging, Visualization and Security

LECTURE (1 HR)	**PROJECT TASK (2 HRS)**
Data storage techniques, dashboards, visualization tools, and IoT security fundamentals.	Create a cloud dashboard for data visualization and implement basic authentication and security mechanisms.

UNIT – IV: INTELLIGENT DECISION LAYER (Weeks 10–12)

Students move from raw sensor data to meaningful decisions by applying filters, building rule-based engines, and deploying lightweight ML models directly on the microcontroller. The unit ends with practical optimization techniques — sleep modes, interrupt-driven logic, and profiling — to make intelligence affordable in constrained hardware.	**Authentic intelligence:** AI tools can scaffold a TensorFlow Lite model deployment in minutes, but they cannot verify whether the model was trained on data representative of your specific sensor, environment, or user population. Before relying on any AI-assisted ML pipeline, document: what data was the model trained on, who is underrepresented, and what happens when the model is wrong — a silent misclassification in a health or safety system is not a software bug, it is a design failure.

Week 10: Data Processing and Filtering

LECTURE (1 HR)	**PROJECT TASK (2 HRS)**
Signal filtering, noise reduction, preprocessing techniques, and feature extraction.	Collect sensor data and apply filtering techniques to improve data quality and reliability.

Week 10: Data Processing and Filtering

Week 11: Rule-Based Systems and Lightweight ML Models

LECTURE (1 HR)	**PROJECT TASK (2 HRS)**
Decision rules, threshold logic, lightweight machine learning models for embedded systems.	Develop a rule-based automation system or implement a lightweight ML model for prediction or classification.

Week 12: Event-Driven Design and Optimization

LECTURE (1 HR) Event-driven architectures, optimization techniques, and performance improvement strategies.	**PROJECT TASK (2 HRS)** Design and implement an event-driven smart system and evaluate optimization approaches.
UNIT – V: CAPSTONE DESIGN PROJECT (Weeks 13–15)	
Students integrate everything from Units I–IV into a complete working prototype — sensing, embedded logic, connectivity, and intelligent decision-making — chosen from five real-world domains including energy, health, agriculture, EV, and safety. The unit culminates in a live demo and a critical reflection on what the system does well, where it could fail, and who it might affect.	**Authentic intelligence:** Deployment is a moral act. Before going live, every team must answer: who could be harmed if this system fails or produces a biased output, and is there a human fallback when it does? AI tools used throughout the project must be named in the final report along with at least two instances where their output was incorrect, misleading, or incomplete — and what human judgment corrected the record.
Week 13: Smart Energy Monitoring & Healthcare Systems	
LECTURE (1 HR) Design considerations, architecture, and implementation strategies.	**PROJECT TASK (2 HRS)** Design and develop a prototype for Smart Energy Monitoring or IoT-based Healthcare Monitoring.
Week 14: Smart Agriculture & Intelligent Security Systems	
LECTURE (1 HR) Sensor deployment, communication architecture, and automation techniques.	**PROJECT TASK (2 HRS)** Develop a smart agriculture or intelligent security system prototype integrating sensing and communication modules.
Week 15: EV Charging Optimization and Project Integration	
LECTURE (1 HR) System integration, testing, validation, and project presentation methodologies.	**PROJECT TASK (2 HRS)** Integrate all system components, validate performance, prepare project documentation, and demonstrate the final solution.

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy):

Use AI-based simulators (e.g., intelligent robotics environments) to classify robot types and configurations.

Basic threshold learning using data trends, calibration using simple regression models.
Sensor fusion using basic statistical models, Fault detection using anomaly thresholds.
Cloud-based predictive analytics, Data-driven visualization insights.
Lightweight ML models (Decision Trees, KNN), Event-driven AI systems.
End-to-end intelligent CPS system, Adaptive and self-learning systems.
Understanding physical-world uncertainty vs digital assumptions Human intuition vs sensor-based decisions.
Trade-off between cost, accuracy, and reliability, Human debugging vs automated diagnostics.
Privacy vs utility in connected systems, Human trust in automated cloud systems.

Explainability vs accuracy in AI decisions, Ethical implications of automation.
Societal impact of deployed systems, Sustainability and ethical responsibility.
AI Tools:
GitHub Copilot, ChatGPT / AI assistants

Applications:

1. IoT-based health monitoring system
2. Smart agriculture (soil moisture, irrigation control)
3. EV charging optimization system
4. Intelligent safety/security system.

Mini Project Component:

- Some sample projects described below simulate the **real software engineering problems**.

Project 1: Develop a sensor data anomaly detection model using real-time data

Project 2: Build a cloud-connected dashboard with predictive insights

Project 3: Implement a TinyML model on ESP32 for smart decision-making

Project 4 – Multi-sensor smart monitoring system with IoT dashboard

Project 5 – AI-enabled predictive maintenance prototype

Project 6 – Smart energy optimization system using real-time data.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Design basic sensing-actuation embedded systems	L6
2	Integrate multiple sensors with reliable embedded interfaces	L3
3	Develop IoT-enabled systems with cloud connectivity	L3
4	Implement intelligent decision-making using AI models	L3
5	Design and deploy a complete CPS solution for real-world problems	L6

COS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	3	2	2	2	–	–	2	2	–	2	3
CO2	3	3	3	2	3	2	–	–	2	2	–	2	3
CO3	2	3	3	2	3	2	2	2	2	2	2	2	3
CO4	2	3	3	3	3	2	2	2	2	2	2	2	3
CO5	3	3	3	3	3	3	2	2	3	3	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Textbooks

1. Vijay Madisetti, Internet of Things: A Hands-on Approach, 1st Ed., Universities Press, 2014.
2. Edward A. Lee & Sanjit A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd Ed., MIT Press, 2017.

Reference books

1. Rajeev Alur, Principles of Cyber-Physical Systems, 1st Ed., MIT Press, 2015Edward A. Lee & Sanjit A. Seshia – Introduction to Embedded Systems

E resources:

1. IEEE research articles

Activity Based Learning (Suggested Activities in Class)

- Sensor Exploration and Interfacing Activity
- Hardware–Software Integration Mini Project
- IoT-Based Data Logging and Monitoring Activity
- AI-Based Decision-Making Robot Activity
- Real-World CPS Problem Design Challenge

****

Data Science and Visualization

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop strong foundations in data science, statistical analysis, and visualization techniques for extracting meaningful insights from complex datasets. It enables students to apply predictive modelling, data wrangling, transformation, and dimensionality reduction techniques using modern Python-based tools and libraries. The course also equips learners with analytical and visualization skills required to interpret, communicate, and support data-driven decision-making through effective graphical representations and intelligent data analysis methods.

Prerequisites: Python Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. Understand and implement predictive modelling approaches using appropriate Python libraries.
2. Employ descriptive statistical methods to explore and prepare datasets effectively.
3. Design and apply suitable data wrangling techniques to handle and refine raw data.
4. Evaluate data from various sources to identify and develop optimal analytical solutions.
5. Apply data transformation and dimensionality reduction techniques to enhance data analysis and modelling.

Broad Course Description

Data Science and Visualization is an interdisciplinary field that combines statistics, programming, and domain knowledge to extract meaningful insights from structured and unstructured data. It involves the entire data lifecycle—data collection, cleaning, transformation, analysis, and interpretation—to support decision-making. Core concepts include predictive modeling, statistical analysis, and machine learning, where techniques are used to identify patterns, trends, and relationships within data. Tools and libraries such as Python-based ecosystems enable efficient handling of large datasets and implementation of analytical models.

Data visualization plays a crucial role in data science by transforming complex data into intuitive graphical representations such as charts, graphs, and dashboards. Visualization techniques help in exploring data (exploratory data analysis), identifying patterns, detecting anomalies, and communicating insights effectively to both technical and non-technical audiences. Methods like histograms, box plots, scatter plots, and heatmaps provide visual clarity and support better understanding of statistical properties like distribution, correlation, and variance.

UNIT – I	06 Hours

Basic Concepts And Python Package

Basic Concepts: Predictive Modelling, Data preparation, Importance of Data preparation, Data Cleaning, Feature selection, Data Transform, Dimensionality reduction, K-fold cross validation, Data Leakage and avoidance measure. Python Packages: NumPy, Matplotlib, pandas, SciPy, scikit, Dataframe, Loading Machine Learning data.

Textbook 1: Chapter 1-7.

UNIT – II	06 Hours

Descriptive Statistics And Data Preparation

Descriptive Statistics: Reasons to study Statistics, Sampling, Data Analysis Process, Mean, Median, Standard Deviation, Skewness, Kurtosis, Graphical Representation-Box Plots, Pivot Table, Heat Map Correlation, Statistics–ANOVA. Data Preparation: Need for Data Pre-processing, Data Transforms, and Rescale Data Standardize Data, Normalize Data, Binarize Data, Univariate and Bivariate Data, Recursive Feature Elimination, Principal Component Analysis.

Textbook 2: Chapter 1, 2, 3, 10.

UNIT – III	06 Hours
Data Cleaning and Feature Selection Data Cleaning: Basic data cleaning, Outlier Identification and Removal, how to Mark and Remove Missing Data, Statistical Imputation, KNN Imputation, Iterative Imputation. Feature Selection: Statistics for feature selection, Methods for categorical input, Methods for Numerical input, Select Features for Numerical Output, RFE for Feature Selection, Significance of feature selection. Textbook 1: Chapter 2, 3, 8.	

UNIT – IV	06 Hours
Data Transform and Dimensionality Reduction: Data Transforms: Scaling data source, minmax scalar and standard scaler, Scale data with outliers, encode categorical data, Make Distributions More Gaussian, Approach for Numerical Data Distributions, Deriving new input variables. Dimensionality reduction: Techniques for Dimensionality Reduction, Linear Discriminant Analysis, PCA Dimensionality Reduction, SVD Dimensionality Reduction Textbook 1: Chapter 4, 5.	

UNIT – V	06 Hours
Other Transforms Transform numerical to categorical, Transform Numerical and Categorical Data, Transform the Target in Regression, Save and load the transformation, case studies for Binary classification, Multiclassification and Regression. Textbook 1: Chapter: 4, 9, 10.	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

AI-Assisted Learning with Critical Reflection: Students use AI tools (e.g., code generators, AutoML platforms) to build ML models, but must critically reflect on outputs, assumptions, and limitations to ensure genuine understanding.

- **Ethical AI and Bias Awareness:** Encourages identification of bias in datasets and models, promoting fairness, accountability, and transparency in ML Applications.
- **Integration of AI Tools with Core Concepts:** AI tools are used as support systems (not replacements), ensuring students still learn algorithms like regression, classification, and clustering fundamentally.
- **Transparency in AI Usage:** Students explicitly document where and how AI tools were used in their ML workflow, ensuring academic integrity and authentic learning.

Applications:

1. Finance & Banking
2. E-Commerce & Retail
3. Transportation & Logistics
4. Healthcare & Medical Diagnosis
5. Education

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the **real software engineering problems**. **Project 1 – Sales Forecasting System**

- **Problem:** Forecast future sales based on historical sales data and trends.
- **Concepts used:** Time Series Analysis, Regression, Data Transformation, Visualization..
- **Industry relevance:** Retail and supply chain management.

Project 2 – Sentiment Analysis on Social Media

- **Problem:** Analyze user sentiments (positive/negative) from social media text data.
- **Concepts used:** NLP, Text Preprocessing, Classification, Visualization.
- **Industry relevance:** Marketing, brand monitoring, and public opinion analysis.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Illustrate predictive modeling techniques with necessary python packages.	L2
CO2	Apply descriptive statistics concepts for data preparation.	L3
CO3	Develop and examine appropriate methods for data wrangling.	L3
CO4	Analyze efficient solution for the given data sources	L4
CO5	Examine the data transformation and dimension reduction techniques on the data source.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO 1	3	2	3		3						2	3	2
CO 2	3	2	2		3						2	3	2
CO 3	3	2	2		3						2	3	2
CO 4	3	2	2		3						2	3	2
CO 5	2	2	2		3						2	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Suggested Tools:

- Development Environments: Jupyter Notebook / Google Colab.
- Python Libraries: NumPy, Pandas, Matplotlib, Seaborn, SciPy, Sckit-learn.
- Data Visualization Tools: Tableau, Microsoft Power BI, Plotly.

TEXT BOOKS:

1. Jason Brownlee, “Data Preparation for Machine Learning” 2020.
2. Roxy Peck, Chris Olsen and Jay Devore, “Introduction to Statistics & Data Analysis ”3rd Edition Thomson Higher Education,2015. ISBN: 1305445961, 9781305445963.

REFERENCE BOOKS:

1. Andrew Park ,“Data Science For Beginners”,2021. ISBN: 9781914167997.
2. Norman Matloff, “Probability and Statistics for Data Science”, CRC Press,2019. ISBN: 9780429687112.

E-RESOURCES:

1. Data Science for Engineers:
   https://digimat.in/nptel/courses/video/106106179/L01.html
2. Statistics for Data Science: https://www.youtube.com/watch?v=V5fqShLVpoI

Activity Based Learning (Suggested Activities in Class)

1. Student seminars (on topics of the syllabus and related aspects (individual activity).
2. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams).
3. Study projects (by very small groups of students on selected local real-time problems about the syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity).

****

Cyber Security Fundamentals

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop strong foundational knowledge in cybersecurity principles, threat analysis, and security mechanisms for protecting digital systems and information assets. It enables students to understand network, system, and application security concepts along with authentication, access control, and risk management strategies. The course also prepares learners to address modern cyber threats and apply AI-enabled monitoring techniques for building secure and resilient computing environments.

Course Learning Objectives:

This Course will enable students to:

1. Understand security principles, CIA triad, and risk management.
2. Recognize threat landscapes—malware, phishing, DoS.
3. Explain network, system, and application security layers.
4. Illustrate authentication, authorization, and access control models.
5. Relate cybersecurity foundations to modern AI-enabled monitoring.

Broad Course Description

Introduces core principles of cybersecurity. Topics include threat taxonomy, security controls, firewalls, crypto basics, and incident response. It lays the groundwork for higher courses on cryptography, IoT security, and blockchain.

UNIT – I	06 Hours

INTRODUCTION TO CYBER SECURITY:

Information security fundamentals – CIA triad, authenticity, non-repudiation.
(Text Book 1 : Ch. 1 1.1–1.5)
Threats, vulnerabilities, exploits, and attack vectors. Security policies, standards (ISO 27001, NIST). Cyber ethics, legal frameworks (IT Act 2000/Amendments). Cyber warfare and critical infrastructure protection. Security architecture layers – physical, logical, organizational.
(Text Book 2 : Ch. 2 2.1–2.3)
Risk management process – identification, analysis, control.
(Text Book 1 : Ch. 3 3.1–3.3)
Case Study: Analysis of recent ransomware incident.

UNIT – II	06 Hours

NETWORK SECURITY & PROTOCOL PROTECTION:

OSI vs TCP/IP models – security implications. Common network attacks: sniffing, spoofing, DoS, DDoS, MITM. (Text Book 2 : Ch. 6–7)
Firewalls: packet-filtering, stateful inspection, proxy and next-gen firewalls.
Intrusion Detection and Prevention Systems (IDS/IPS) – architectures and signatures. Secure network protocols – HTTPS, SSH, TLS, VPNs. Wireless security standards – 802.11i/WPA 2/3, EAP. Network segmentation and zero-trust architecture basics.
(Text Book 1 : Ch. 4-7) (Text Book 2 : Ch. 7-10)
Case Study: Analysis of OWASP Top 10 vulnerabilities.

UNIT – III	06 Hours

SYSTEM & APPLICATION SECURITY:

OS security concepts – kernel hardening, patch management. Access control models (DAC, MAC, RBAC, ABAC). Malware taxonomy – virus, worm, ransomware, spyware. Application vulnerabilities – buffer

overflow, SQL injection, XSS, CSRF. Secure coding principles and SDLC integration (DevSecOps). Mobile and cloud application security. Virtualization and container security concepts (Docker hardening). (Text Book 1 : Ch. 8-12) (Text Book 2 : Ch. 11-13)	
UNIT – IV	**06 Hours**
INCIDENT MANAGEMENT & FORENSICS: Incident response lifecycle – preparation, detection, containment, recovery. Log management and SIEM systems (Splunk, ELK stack). Digital forensics – data acquisition, preservation, examination, reporting. Forensic imaging tools (FTK Imager, Autopsy). Legal admissibility of digital evidence. Disaster recovery and business continuity planning. Security metrics and Key Risk Indicators (KRI). (Text Book 1 : Ch. 14-18) (Text Book 2 : Ch. 14-16)	
UNIT – V	**06 Hours**
EMERGING TRENDS AND AI-DRIVEN CYBER DEFENSE: Cloud security models (SaaS, PaaS, IaaS). Identity and Access Management (IAM) with Azure AD / Okta. Secure configuration management and compliance automation. AI in cybersecurity – ML for anomaly detection, phishing identification. Ethical and privacy implications of AI-based defense. Quantum computing impact on cybersecurity. Security in IoT and industrial systems. (Text Book 1 : Ch. 19-25) (Text Book 2 : Ch. 17-18) **Case Study:** AI vs AI attacks (adversarial ML).	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Sustainable, ethical monitoring practices.
- Automation in threat detection using AI models.
- AI-driven threat detection and traffic classification.
- Automated static code analysis using ML classifiers.
- Predictive forensics and threat prioritization using AI

Applications

1. Enterprise risk management and compliance using CIA triad, ISO 27001, and NIST standards.
2. Network defense through firewalls, IDS/IPS, VPNs, and zero-trust architecture.
3. Secure software development with DevSecOps and protection against OWASP Top 10 vulnerabilities.
4. Incident response and digital forensics using SIEM tools, evidence preservation, and recovery planning.
5. AI-driven cyber defense for anomaly detection, phishing prevention, IoT security, and quantum-resistant cryptography.

Mini Project Component

Project 1 – AI-Driven Intrusion Detection System (IDS)

- **Problem:** Configure a simple IDS using Snort/Wireshark and integrate AI² for real-time anomaly detection and traffic classification.**Concepts used:** Packet sniffing, rule-based detection, machine learning classifiers, anomaly detection.
- **Industry relevance:** Network security monitoring and AI-powered SOC platforms (Google Cloud Security, Amazon GuardDuty).

Project 2 – Automated Static Code Analysis for OWASP Top 10

- **Problem:** Perform automated static code analysis using AI² to detect OWASP

Top 10 vulnerabilities in Applications.

- **Concepts used:** Secure coding practices, ML-based code parsing, vulnerability classification, DevSecOps automation.
- **Industry relevance:** Application security audits and secure DevOps pipelines (GitHub Advanced Security, AWS CodeGuru).

LAB EXERCISES

1. Installation and configuration of firewalls and packet analyzers.
2. Malware behavior analysis using sandboxing tools.
3. Identify SQL Injection and implement secure input validation.
4. GIAC-style capture-the-flag (CtF) simulation.
5. Report on forensic log analysis for breach incident.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain and apply fundamental cybersecurity concepts and policies.	**L2**
CO2	Identify and analyze network/system vulnerabilities.	**L3**
CO3	Demonstrate security mechanisms and mitigation strategies.	**L3**
CO4	Investigate security incidents using forensic approaches.	**L3**
CO5	Employ AI tools for intelligent cyber defense in emerging environments.	**L4**

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2										2	
CO2	2	3	3		2							3	
CO3	2	2	3	2	2							2	
CO4	3	2										2	
CO5	2	2	3		3							3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. William Stallings – *Computer Security: Principles and Practice*, 8e, Pearson, 2023.
2. Mark Ciampa – *Security+ Guide to Network Security Fundamentals*, 7e, Cengage, 2022

REFERENCE BOOKS:

1. Charles Pfleeger and Shari Lawrence Pfleeger – *Security in Computing*, 5e.
2. Matt Bishop – *Computer Security Art and Science*, 2e, Addison Wesley.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Fundamentals of Cybersecurity Simulations w ith Morpheus	II, V	AI-driven security and threat analyti cs
Deep Learning for Anomaly Detection	V	Behavioral analytics and intrusion patterns
Edge AI with Jetson	V	IoT security gateway intelligence

E-RESOURCES:

1. nptel.ac.in
2. coursera.org
3. edx.org
4. cybrary.it
5. udemy.com
6. developer.nvidia.com
7. cse29-iiith.vlabs.ac.in

Activity-Based Learning (Suggested Activities in Class)

1. Case Study Discussion: Analyze a recent cyber-attack (ransomware or data breach) and identify weaknesses and preventive controls.
2. Threat Modeling Exercise: Students design a simple network architecture and mark pot ential entry points and countermeasures.
3. Hands-on Demo: Capture and inspect network traffic using Wireshark to detect anomal ies or malicious packets.
4. Role Play (SOC Simulation): Groups act as Security Operations Center analysts respon ding to a simulated incident.
5. Ethical Debate: Discuss “AI in Cybersecurity – Balancing Privacy and Protection” to promote responsible security thinking.

****

Interactive Art and Creative Coding

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	04 Hours	**Total Hours**	:	30Hrs L+15 Hrs J
L–T–P-J	:	2–0–0-2	**CIE Marks**	:	60 + 40 Marks

Course Vision:

The course aims to develop creative and technical skills at the intersection of art, design, and programming through interactive and generative digital media. It enables students to create expressive visual, audio, and multimodal experiences using creative coding frameworks, interactive technologies, and AI-assisted artistic tools. The course also fosters innovation, computational creativity, and artistic experimentation while equipping learners to design publishable digital artworks and immersive interactive experiences for modern creative industries.

Course Learning Objectives:

This Course will enable students to:

1. Explore the intersection of art, design, and programming through computational creativity.
2. Develop skills in creative coding using Processing, p5.js, and generative art frameworks.
3. Design interactive audiovisual experiences using code-driven animation and multimodal user interaction.
4. Understand generative art principles and apply them to create expressive, publishable digital artworks.

Broad Course Description

Interactive Art and Creative Coding fuses programming with visual art, design, music, and human-computer interaction. Using Processing and p5.js as primary environments, students create dynamic visual compositions through generative patterns, particle systems, Perlin noise, physics simulations, and algorithmic design. The course covers interactive design with mouse, keyboard, camera, and microphone inputs, AI-based interactions using ml5.js (PoseNet, FaceAPI), sound synthesis with Tone.js, 3D graphics via Three.js, and AI art generation tools (Stable Diffusion, RunwayML). Students build a portfolio of published creative coding projects demonstrating both technical skill and artistic vision.

UNIT – I	**06 Hours**

Foundations of Creative Coding: Introduction to creative coding – history and movements (Casey Reas, Vera Molnár, John Maeda). Processing IDE and p5.js environment setup. Drawing primitives – shapes, colors, coordinate system. Typography and image manipulation in code. Variables, conditionals, loops in a creative context. Functions for reusable generative components. Randomness – random() and noise() for controlled unpredictability. (Ref: Casey Reas & Ben Fry, Getting Started with Processing Ch. 1–4)

Textbook Reference:

• Casey Reas & Ben Fry, Getting Started with Processing, 2nd Ed – Ch. 1–4 (Maker Media, 2015)
• p5.js Reference Documentation (p5js.org/reference)

UNIT – II	**06 Hours**

Animation & Motion Systems: Animation principles in code – frame rate, easing, interpolation. Object-Oriented approach for animated entities. Particle systems – creation, emission, and lifecycle. Physics simulation – gravity, friction, spring forces. Trigonometric motion – circular, pendulum, wave animations. PVector/createVector mathematics for natural movement. Perlin noise for organic motion, terrain, and landscapes. (Ref: Shiffman, The Nature of Code Ch. 1–4)

Textbook Reference:

• Daniel Shiffman, The Nature of Code – Ch. 1–4 (natureofcode.com, 2024)
• Casey Reas & Ben Fry – Ch. 5–7

UNIT – III	06 Hours
Interactivity & Input Devices: Mouse and keyboard interaction – events, states, and gestures. Touch interaction for mobile p5.js sketches. Camera input and computer vision with ml5.js – PoseNet, FaceAPI, BodyPix. Microphone input – audio analysis, FFT, frequency visualization. Webcam-based interactive installations. Game controller and MIDI input integration. Building interactive web-based art experiences with A-Frame basics. (Ref: Shiffman, The Nature of Code Ch. 8–9) Textbook Reference: • Daniel Shiffman, The Nature of Code – Ch. 8–9 • ml5.js Documentation (ml5js.org)	
UNIT – IV	**06 Hours**
Generative Art & Algorithmic Design: Generative systems – L-Systems, Fractal geometry (Mandelbrot, Julia sets). Cellular Automata – Conway's Game of Life and rule-based systems. Algorithmic patterns – Voronoi diagrams, Truchet tiles, flow fields. Data-driven art – visualizing real datasets creatively. 3D graphics in Processing and WebGL with p5.js. Three.js basics for 3D web art. Introduction to GLSL shaders for GPU-accelerated visuals. (Ref: Matt Pearson, Generative Art Ch. 1–5) Textbook Reference: • Matt Pearson, Generative Art: A Practical Guide using Processing – Ch. 1–5 (Manning, 2011) • Daniel Shiffman, The Nature of Code – Ch. 5–7	
UNIT – V	**06 Hours**
Sound, AI Art & Exhibition: Sound synthesis and music programming with Tone.js. Audiovisual synchronization – music-reactive visuals using FFT. Introduction to AI-generated art – DALL-E, Stable Diffusion, Runway ML. Integrating AI-generated assets into interactive sketches. Creative workflow – ideation, prototyping, refining, and critique. Publishing web-based art to GitHub Pages and OpenProcessing. Ethics of AI art – authorship, originality, attribution, and copyright. (Ref: Tone.js Documentation) Textbook Reference: Tone.js Official Documentation (tonejs.github.io), Runway ML Documentation (runwayml.com/docs)	

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Unit 5	**Generative AI Explained**	Generative AI	1 Hr	—	courses.nvidia.com/course-v1:DLI+S-FX-07+V1	Essential conceptual foundation for AI art tools (DALL-E, Stable Diffusion, Runway ML) covered in Unit 5 — students must understand how generative AI works before critically using, integrating, and evaluating AI art tools.

Unit 5	**Generative AI With Diffusion Models**	Generative AI / Image	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-v1:DLI+T-FX-11+V1	Deep dive into diffusion model-based image generation (DDPM, score matching, DALL-E concepts) — directly supports Unit 5's Stable Diffusion and AI art generation module with hands-on
						implementation context.
Unit 3	**Building a Brain in 10 Minutes**	Deep Learning	10 Min	—	courses.nvidia.com/course-v1:DLI+T-FX-01+V1	10-minute neural network introduction — bridges the gap for art students encountering ml5.js AI (PoseNet, FaceAPI) in Unit 3, providing conceptual grounding for how AI interprets body and face inputs.
Units 3–4	**Getting Started With Deep Learning**	Deep Learning	2 Hrs	—	courses.nvidia.com/course-v1:DLI+S-FX-01+V1	Introductory deep learning course with TensorFlow/Keras and CNNs — foundational for students wanting to understand how ml5.js models work under the hood and how to fine-tune them for creative installations.
Unit 4	**Synthetic Tabular Data Generation Using Transformers**	Generative AI	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-v1:DLI+T-FX-13+V1	Synthetic data generation using Transformer models — connects to Unit 4's data-driven art module by showing how AI can generate structured datasets that can be visualized and sonified creatively.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

• **AI as Creative Collaborator:** Use DALL-E and Midjourney as inspiration sources and integrate AI-generated assets into interactive p5.js sketches.

• **Reflective Authorship:** Compare AI-generated art with student code-generated work; critically examine creativity, intent, originality, and attribution.

• **Data-Driven Storytelling:** Visualize real datasets creatively with p5.js; reflect on how data selection and framing shapes narrative and perception.

• **Computational Aesthetics:** Discuss what makes algorithmic art beautiful; analyze how human aesthetic judgment guides the development and evaluation of AI art outputs.

• **Interactive AI Experiences:** Build web experiences using ml5.js PoseNet responding to body movement; reflect on privacy, consent, and bodily data in interactive installations.

Applications:

- Digital Art Installations and Museum Exhibitions
- Music Visualization and Live Audiovisual Performance
- Interactive Brand Experiences and Advertising
- Data Visualization and Infographic Design
- Game UI/UX Prototyping and Motion Design

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Create a generative poster using loops, randomness, and algorithmic color palettes in p5.js.
- Build an animated particle system responding to mouse movement with simulated gravity and friction.
- Implement a Perlin noise-based landscape/terrain generator with animated camera movement.
- Create an interactive kaleidoscope pattern responding to mouse position and click events.
- Build a music visualizer reacting to microphone input using FFT frequency band analysis.
- Implement a PoseNet-based sketch where user body movement controls visual elements in real time.
- Design a Cellular Automata simulation (Conway's Game of Life) with interactive rule modification.
- Create a fractal explorer (Mandelbrot/Julia set) with zoom, color mapping, and parameter controls.
- Build a 3D rotating abstract sculpture using Processing 3D mode or WebGL in p5.js.
- Design and publish a complete interactive web art experience on OpenProcessing or GitHub Pages.

Mini Project Component

• **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – Generative Art Series

Problem: Five algorithmically generated artworks with a unifying theme, each with distinct interactive elements.

Concepts used: p5.js, Generative algorithms, Color theory, Randomness

Industry relevance: Digital art galleries, NFT platforms, design portfolios.

Project 2 – Music-Reactive Visual Performance

Problem: Real-time audiovisual performance tool generating synchronized visuals from live or recorded music.

Concepts used: Tone.js, FFT analysis, Particle systems, Timing

Industry relevance: Live performance, music production, entertainment.

Project 3 – Body-Tracking Interactive Installation
Problem: Webcam captures body movement via PoseNet, triggering generative visual and sound outputs.

Concepts used: ml5.js PoseNet, Interactive design, Sound synthesis

Industry relevance: Museum installations, interactive advertising, therapy.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply fundamental creative coding concepts to create expressive visual compositions using Processing and p5.js.	L3
2	Design and implement animated motion systems using physics simulation, particle effects, and trigonometric motion.	L3
3	Build interactive experiences that respond to user input from mouse, keyboard, camera, and microphone inputs.	L3
4	Create generative art systems using algorithmic design principles such as fractals, L-systems, and cellular automata.	L4
5	Integrate AI art tools and sound synthesis into creative coding projects and present polished interactive artworks.	L5

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	2	2	3	2	3						2	2	3
CO2	2	3	2	3	3						2	2	3
CO3	3	2	3	3	3						2	3	3
CO4	3	3	3	3	2						2	3	2
CO5	2	3	3	3	3						2	3	3

Suggested Tools:

- Processing 4.x – Creative Coding IDE
- p5.js – JavaScript Creative Library
- ml5.js – Friendly Machine Learning for Web
- Tone.js – Web Audio Framework
- Three.js – 3D JavaScript Library
- OpenProcessing.org – Share and Explore Sketches
-

TEXT BOOKS:

1. Daniel Shiffman, The Nature of Code, natureofcode.com, 2024.
2. Casey Reas & Ben Fry, Getting Started with Processing, 2nd Ed, Maker Media, 2015.
3. Matt Pearson, Generative Art: A Practical Guide using Processing, Manning, 2011.

REFERENCE BOOKS:

1. Philip Galanter, What is Generative Art?, NYU Research Paper, 2003.
2. John Maeda, Design by Numbers, MIT Press, 2001.

E-RESOURCES:

- https://thecodingtrain.com (The Coding Train by Daniel Shiffman)
- https://openprocessing.org
- https://p5js.org
- https://learn.nvidia.com (NVIDIA DLI Free Courses)

Activity Based Learning (Suggested Activities in Class)

- Weekly creative challenges – 'Sketch of the Week' shared on class gallery board.
- Collaborative generative art exhibition displayed publicly at end of semester.
- Art critique sessions using design vocabulary and aesthetic evaluation rubrics.
- Creative coding live demo: instructor codes a generative sketch from scratch in class.

****

Introduction to Quantum Computing

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 hrs T
L–T–P-J	:	3–0–0-0			

Prerequisites: Python Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

1. Understand the mathematical foundations required for quantum computing and quantum information theory.
2. Analyze qubits, quantum states, vector spaces, matrices, and operators used in quantum systems.
3. Apply principles of quantum mechanics, tensor products, and quantum state representations.
4. Design and analyze basic quantum gates and quantum circuits.
5. Understand quantum measurement, entanglement, and their applications in modern quantum technologies.

COURSE VISION:

To equip students with foundational knowledge of quantum mechanics and computational principles for solving complex problems.

To inspire innovation by applying quantum algorithms and circuits in engineering, science, and emerging technologies.

Broad Course Description

The course "Introduction to Quantum Computing" provides a comprehensive foundation in the mathematical and computational principles underlying quantum computation and quantum information processing. The course introduces learners to the concepts of qubits, quantum states, vector spaces, operators, tensor products, quantum gates, quantum circuits, quantum measurement, and entanglement. It bridges the gap between classical computation and emerging quantum technologies by combining mathematical rigor with computational intuition. Students will gain knowledge of the theoretical foundations of quantum mechanics relevant to computing and develop the ability to analyze simple quantum circuits and quantum algorithms. The course also emphasizes practical exposure to quantum programming environments and emerging applications of quantum computing in artificial intelligence, cryptography, optimization, and communication systems.

UNIT – I	09 Hours

Mathematical Foundations of Quantum Computing, Qubits, and quantum states

Classical Information, Information Content in a Signal, Entropy and Shannon's Information Theory, Probability Basics. The Qubit, Vector Spaces, Linear Combinations of Vectors, Uniqueness of a Spanning Set, Basis and Dimension, Inner Product, Orthonormality, Gram-Schmidt Orthogonalization, Bra-Ket Formalism, The Cauchy-Schwartz and Triangle Inequalities.

(Text Book-1: Chapter 1, Chapter 2)

UNIT – II	10 Hours

Matrices & Operators - Pauli operators, outer products, Representations of Operators Using Matrices, outer products & matrix representation, Matrix Representation of Operators in Two-Dimensional Spaces, The Pauli Matrices, Hermitian, unitary & normal operators, eigenvalues and eigen vectors, characteristic equation, trace of an operator, expectation value of an operator, projection operators. (Text Book-1: Chapter 3)	
UNIT – III	**08 Hours**
Postulates and Tensor Products The Heisenberg Uncertainty Principle, Polar Decomposition and Singular Values, The Postulates of Quantum Mechanics, Representing Composite States in Quantum Mechanics, Computing Inner Products, Tensor Products of Column Vectors, Operators and Tensor Products, Tensor Products of Matrices. (Text Book1: Chapter 3, Chapter 4)	
UNIT – IV	**09 Hours**
Quantum Gates and Circuits: classical logic gates and circuits, single qubit quantum gates, the Hadamard gate, two qubit quantum gates- the CNOT gate, three qubit quantum gates- The Fredkin gate, The Toffoli gate, quantum circuits, universal quantum gates. (Text Book1: Chapter 8)	
UNIT – V	**09 Hours**
Quantum Measurement and Entanglement: Distinguishing Quantum States and Measurement, Projective Measurements, Measurements on Composite Systems, Generalized Measurements, Positive Operator-Valued Measures, Entanglement: Bell's Theorem, Bipartite Systems and the Bell Basis, When Is a State Entangled?, The Pauli Representation, Entanglement Fidelity, Using Bell States For Density Operator Representation, Schmidt Decomposition (Text Book1: Chapter 6, Chapter 7)	

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI–Quantum Computing Foundations:** Integrate fundamental concepts of artificial intelligence and quantum computing to understand how quantum principles can enhance intelligent computational systems and future AI applications.
- **Quantum Mathematical Intelligence:** Apply linear algebra, vector spaces, matrices, tensor products, and probability theory to model quantum states and computational processes required for intelligent quantum systems.
- **Quantum Information Representation:** Utilize qubits, superposition, and quantum state representations to encode and process information beyond the limitations of classical binary systems.
- **Quantum Circuit-Based Computational Thinking:** Design and analyze quantum gates and quantum circuits to develop logical reasoning and problem-solving abilities for next-generation intelligent computing systems.
- **Simulation-Driven Quantum Learning:** Employ quantum simulation platforms such as Qiskit and IBM Quantum Experience for experimentation, validation, and visualization of quantum operations and circuits.
- **Hybrid Quantum–Classical Learning Environment:** Encourage the integration of classical computational approaches with quantum paradigms to understand emerging hybrid intelligent systems and optimization strategies.

- **Experiential and Inquiry-Based Learning:** Promote active learning through hands-on circuit design, simulation exercises, case studies, and mini projects to strengthen conceptual understanding and innovation skills.
- **Ethical and Responsible Quantum Intelligence:** Develop awareness regarding ethical implications, security considerations, and responsible use of quantum technologies and intelligent computational systems.
- **Research-Oriented Quantum Exploration:** Encourage students to analyze recent advancements in quantum computing, quantum machine learning, and quantum communication through seminars, research reviews, and exploratory activities.
- **Authentic Intelligence Development (AI²):** Foster analytical reasoning, creativity, collaboration, and critical thinking skills to prepare learners for future interdisciplinary research and industry applications in quantum technologies and intelligent systems.

Suggested Tools:

Qiskit
IBM Quantum Lab
PennyLane
Google Cirq

Applications:

Students will explore applications of quantum computing in:

1. Quantum Cryptography
2. Secure Communication Systems
3. Quantum Machine Learning
4. Drug Discovery and Molecular Simulation
5. Financial Modeling and Optimization
6. Natural Language Processing
7. Cybersecurity
8. Supply Chain Optimization
9. Weather Forecasting
10. Smart Healthcare Systems

Mini Project Component

Industry-Driven Projects (IBM / Google Quantum Style)

These projects are designed to simulate real-world quantum computing and intelligent system applications using modern quantum technologies and simulation platforms.

Project 1 – Quantum Gate Simulation System

1. **Problem:** Design and simulate fundamental quantum gates and analyze their behavior on qubit states.
2. **Concepts used:** Qubits, Pauli gates, Hadamard gate, CNOT gate, quantum circuits
3. **Tools/Platforms:** Qiskit
4. **Industry relevance:** Quantum processor design, quantum hardware testing, research laboratories

Project 2 – Quantum Circuit Visualization Platform

1. **Problem:** Develop a system to create and visualize simple quantum circuits for educational and analytical purposes.
2. **Concepts used:** Quantum circuits, tensor products, measurement, quantum states
3. **Tools/Platforms:** IBM Quantum Experience

4. **Industry relevance:** Quantum education technology, simulation platforms, research training

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the mathematical foundations of quantum computing, including qubits, vector spaces, entropy, probability theory, and quantum states.	L2
2	Analyze matrices, operators, eigenvalues, eigenvectors, and quantum state transformations used in quantum systems.	L4
3	Apply tensor products and postulates of quantum mechanics to represent and analyze composite quantum systems.	L3
4	Design and evaluate quantum gates and quantum circuits for solving basic computational and information processing problems.	L5
5	Analyze quantum measurement techniques, entanglement properties, and their applications in quantum information processing systems.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2			2						2	2	
CO2	3	3	2	2	3						2	3	2
CO3	2	3	2	2	3						2	3	2
CO4		2				2	3				2	2	
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. David McMahon, **"Quantum Computing Explained",** IEEE Computer Society, WILEY Interscience
2. Noson S. Yanofsky, Mirco A. Mannucci, "**Quantum Computing for Computer Scientists**", Cambridge University Press

REFERENCE BOOKS:

1. Michael A. Nielsen. & Issac L. Chiang, "Quantum Computation and Quantum Information", South Asia Edition 2013, Cambridge University Press, Reprint 2022.
2. "Quantum Computing, A Gentle Introduction", Eleanor G. Rieffel and Wolfgang H. Polak MIT press (2014).
3. Mikio Nakahara and Tetsuo Ohmi, "Quantum Computing", CRC Press (2008). 3. N. David Mermin, "Quantum Computer Science", Cambridge (2007).
4. An introduction to Quantum Computing, Phillip Kaye, Raymond Laflamme, Muchele Mosca, Oxford University Press, 2007, ISBN-13: 978-0198570493, ISBN-10: 019857049X.

E-Resources:

- https://quantum.ibm.com
- https://qiskit.org
- https://azure.microsoft.com/en-us/products/quantum
- https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Quantum Circuit Design Activity

Activity 2: Mini Project (Quantum random number generator, Quantum password generator, Quantum circuit simulator)

Activity 3: Visualization Activity (Qubit states, Quantum circuits)

****

Product Strategy & Management

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P-J	:	3–0–0-0			

Course Vision:

The course aims to develop strong foundations in product strategy and product management for designing, evaluating, and managing successful products in competitive markets. It enables students to understand customer behavior, product life cycle management, portfolio strategies, and new product development processes for creating sustainable business value. The course also equips learners with analytical, strategic, and decision-making skills required to transform innovative product ideas into commercially successful market solutions.

Course Learning Objectives:

This Course will enable students to:

1. Develop a strong foundation in product strategy concepts, competitive positioning, and product management principles.
2. Enable students to understand and analyze buyer behavior and its influence on product decisions.
3. Provide knowledge of product life cycle and portfolio management for effective product planning.
4. Equip students with the skills to manage the new product development (NPD) process from idea generation to commercialization.
5. Enhance the ability to evaluate product ideas using concept testing, business analysis, and product testing techniques.

Broad Course Description

This course provides a comprehensive understanding of product strategy and management in competitive markets. It explores the theoretical foundations of product concepts, customer behavior, and market dynamics, along with practical approaches to managing product life cycles and portfolios. The course emphasizes the new product development (NPD) process, including idea generation, screening, concept testing, and commercialization decisions. Through strategic frameworks and real-world Applications, students will learn how to design, evaluate, and manage products to achieve sustainable competitive advantage and market success.

UNIT – I	07 Hours

Foundations of Product Strategy

Competition and Product Strategy, The Product in Theory and Practice

UNIT – II	08 Hours

Understanding Markets and Customers

Buyer Behaviour, Market Needs and Customer Insights

UNIT – III	10 Hours

Product Life Cycle and Portfolio Management

Product Life Cycle in Theory and Practice, Innovation and Portfolio Balance

UNIT – IV	10 Hours

New Product Development Process

Importance, Nature, and Management of NPD, New Product Strategy, Idea Management for NPD.

UNIT – V	10 Hours

Product Development and Evaluation
Screening New Product Ideas, Concept Development and Testing, Business Analysis, Product Testing

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)
AI-Enabled Product Ideation & Concept Development:

Use AI tools to generate product ideas, refine concepts, and align them with market needs while ensuring originality and contextual relevance.

• AI-Assisted Market & Customer Analysis:

Leverage AI to analyze customer behaviour, segment markets, and derive insights for product positioning, while critically evaluating the accuracy and bias of AI outputs.

• Responsible & Ethical Use of AI in Product Management:

Ensure ethical use of AI by validating outputs, avoiding over-dependence, maintaining transparency, and reflecting on biases and limitations in AI-supported decision-making.

Suggested Tools:

- Miro
- Figma

Applications:

- Business analysis, forecasting, product performance evaluation
- Product presentations, concept visualization
- Product prototyping and interface design

Project Component

- Identify an unmet customer need in a specific market segment and develop a new product concept, including idea generation, screening, concept testing, and business analysis.
- Analyze the product portfolio of a company and recommend strategies for resource allocation, product addition, or elimination to maximize profitability and growth.
- Use AI tools to generate and refine a product idea, and evaluate its feasibility, customer value, and ethical considerations.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the concepts of product strategy, competition, and product management.	L2
2	Analyze buyer behaviour and its impact on product decisions.	L4
3	Evaluate product life cycle stages and portfolio strategies.	L5
4	Apply new product development processes and strategies.	L3
5	Assess product ideas using business analysis, testing, and evaluation techniques.	L4

Table: Mapping Levels of COs to POs / PSOs												
COS	Program Outcomes (POs)										Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	2	3			2	2	2	2		2
CO2	3	3	2	2	3			2	2	2	2		2
CO3	2	3	3	2	3			2	2	2	2		2
CO4	2	2	3	3	3			2	2	2	2		2
CO5	3	3	2	2	3			2	2	2	2		2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Michael Baker and Susan Hart, Product Strategy and Management, Pearson Education, 2nd Edition, 2008. ISBN – 9788131724286

REFERENCE BOOKS:

1. Philip Kotler and Kevin Lane Keller, Marketing Management, Pearson, 14th Edition, 2013.
2. Donald Lehmann and Russell Winer, Product Management, McGraw-Hill, 4th Edition, 2005.
3. Marc Annacchino, New Product Development: From Initial Idea to Product Management, Elsevier, 2003.

E-RESOURCES:

- https://hbr.org/topic/product-management
- https://www.productplan.com/learn/
- https://www.interaction-design.org/literature/topics/product-design

Activity Based Learning (Suggested Activities in Class)

- Identify real-world product failures and analyze reasons using product strategy concepts.
- Develop a new product idea and carry out concept testing with peer/customer feedback.
- Conduct product life cycle analysis for existing products in different industries.
- Design a product portfolio strategy for a company using case-based scenarios.
- Perform market research to understand buyer behavior and product positioning.

****

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Data Analytics and Visualization with MS PowerBI [IIMB] 6 WEEKS Data Analytics and Visualization with MS Power BI | SWAYAM https://onlinecourses.swayam2.ac.in/e-learning/preview/imb26_mg110 Data Analytics with Python [NPTEL] 12 WEEKS Preview: Data Analytics with Python | SWAYAM https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs86
2	*CID	Practical Cyber Security for Cyber Security Practitioners, IIT Kanpur [NPTEL] 12 WEEKS Practical Cyber Security for Cyber Security Practitioners | SWAYAM https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs120
3	*CID	Quantum Computation and Quantum Information [NPTEL] 12 WEEKS Quantum Computation and Quantum Information | SWAYAM https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ph02
4	*CID	Product and Brand Management, IIT Roorkee [NPTEL] 12 WEEKS Product and Brand Management | SWAYAM https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_mg82

Note: CID = NPTEL Course ID

Sustainable Software Engineering

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P-J	:	3 – 0 – 0 – 0			

Course Vision: The course aims to develop environmentally responsible software engineers capable of designing and developing sustainable, energy-efficient, and eco-friendly software systems. It enables students to integrate green computing principles, sustainable coding practices, and carbon-conscious decision-making into modern software development and AI applications. The course also promotes innovation in scalable and sustainable computing solutions while fostering awareness of the environmental impact of IT systems to support global sustainability goals and a greener digital future.

Prerequisites: Software Engineering Fundamentals

Course Vision:

To cultivate environmentally responsible software engineers who design and develop sustainable, energy-efficient computing solutions, integrate green computing principles into modern software development practices, promote awareness of the environmental impact of IT systems and encourage carbon-conscious decision-making, foster innovation in building scalable and eco-friendly software and AI systems, and contribute towards a sustainable digital future by aligning software engineering practices with global environmental goals.

Course Learning Objectives:

Students will :

1. Introduce the fundamentals of Green IT, environmental concerns, and the role of computing in sustainable development.
2. Provide knowledge of green software engineering principles and techniques for developing energy-efficient software systems.
3. Develop skills in writing optimized and energy-efficient code using sustainable programming practices and measurement techniques.
2. Enable understanding of integrating sustainability into all phases of the software development lifecycle and operations.
3. familiarize students with sustainable approaches in AI/ML systems and reduce the environmental impact of intelligent applications.

Broad Course Description

To cultivate environmentally responsible software engineers who design and develop sustainable, energy-efficient computing solutions, integrate green computing principles into modern software development practices, promote awareness of the environmental impact of IT systems and encourage carbon-conscious decision-making, foster innovation in building scalable and eco-friendly software and AI systems, and contribute towards a sustainable digital future by aligning software engineering practices with global environmental goals.

This course introduces the principles of green computing and sustainable software engineering, focusing on reducing the environmental impact of software systems. It covers energy-efficient design, green coding practices, and the role of IT in minimizing energy consumption and carbon emissions through informed development decisions. The course also addresses sustainability across the software lifecycle, including design, deployment, and maintenance, while exploring modern technologies like cloud computing, artificial intelligence, and machine learning. Students gain practical skills to measure and optimize software

energy efficiency, preparing them to develop environmentally responsible and sustainable applications.	
Detailed Course Syllabus The course consists of five modules. Each module integrates: ● Core topics ● Engineering applications ● AI integration ● Authentic Intelligence reflection	
UNIT – I	**09 Hours**
Green IT and Intelligent Sustainable Systems (9 Hours): Introduction- Environmental Concerns and Sustainable Development- Environmental Impacts of IT-Green IT- Holistic Approach to Greening IT-Applying IT for Enhancing Environmental Sustainability- Enterprise Green IT Strategy- Green IT: Burden or Opportunity? - Role of Generative AI (LLMs) and Agentic AI in intelligent energy-aware systems.	
UNIT – II	**09 Hours**
Green Software Engineering and Data-Driven Optimization (9 Hours): Introduction- The problem-More devices require more energy - Data Centre Proliferation - Green Hosting-Green Development-Carbon offsetting- The principles - Measurement and optimization - use of Accelerated Data Science, AI Networking, and Deep Learning for energy optimization - Applying the principles, Case study on carbon aware computing.	
UNIT – III	**10 Hours**
Green Coding Practices and Accelerated Computing (10 Hours): What Is Code Efficiency? - The Benefits of Efficiency - Developer Productivity - Use Green Design Patterns - Carbon Intensity of Electricity - Demand Shifting - Real-World Examples – Measurement - Examination of the Current Methodologies - Available Tooling: Fundamentals of Accelerated Computing with Modern CUDA C/C++, Fundamentals of GPU-Accelerated Workflows with CUDA Python, Case study: Optimizing energy efficiency at data centres.	
UNIT – IV	**08 Hours**
Sustainable Software Engineering Lifecycle and Digital Twin Systems (8 Hours): Operational Efficiency - Machine Utilization - Hyperscalers and Profit – LightSwitchOps – Monitoring - Co-Benefits - The Green Software Maturity Matrix - Integration of Industrial Digital Twin systems and OpenUSD frameworks for simulation and optimization.	
UNIT – V	**09 Hours**

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

AI Integration:

Module-1:

- **Generative AI LLM** supports understanding of large-scale AI system energy consumption, optimization of inference workloads, and sustainable deployment of foundation models.
- **Agentic AI** aligns with intelligent autonomous decision-making for dynamic energy optimization, workload scheduling, and sustainability-aware resource allocation.

Module-2:

- **Accelerated Data Science** enables energy analytics, predictive sustainability modeling, and optimization of large-scale operational data.

- **AI Networking** supports efficient communication architecture to reduce transmission energy overhead.
- **Deep Learning** helps model and predict energy usage patterns for carbon-aware optimization.

Module-3:

- **Accelerated Computing** directly supports GPU-based performance optimization, efficient parallel execution, and reduced compute energy consumption.
- **CUDA training** enables students to write optimized software with lower execution time and power usage.
- **Computer Vision & Video Analytics** demonstrates optimization of high-compute AI workloads for sustainable execution.

Module-4:

- **Industrial Digital Twin** supports simulation-driven sustainability testing and operational optimization before physical deployment. **OpenUSD** enables scalable modeling of system interactions for lifecycle efficiency analysis.

Module-5:

- **Deep Learning** enables efficient model architecture design and training optimization.
- **Generative AI LLM** supports sustainable large-model deployment strategies.
- **Computer Vision & Video Analytics** aligns with optimization of compute-intensive inference pipelines.
- **Agentic AI** enables autonomous adaptation for energy-efficient AI operation.

Authentic Intelligence Reflection:

Module-1: Encourages students to critically evaluate whether technology growth aligns with long-term environmental sustainability goals.

Module-2: Develops awareness of balancing innovation with responsibility in designing energy-efficient software systems.

Module-3: Promotes mindful coding habits where developers take responsibility for the environmental impact of their code.

Module-4: Encourages thinking beyond development to lifecycle accountability and continuous sustainability improvement.

Module-5: Builds ethical awareness about the environmental cost of AI and promotes responsible AI development practices.

Assessment Scheme

Definitions

1. **"Internal Assessment"** refers to the Continuous Internal Assessment (CIA). The weightage of this component is **60%**. It includes:

   a. One **Mid Semester Examination (MSE)** (Descriptive-based),

   b. **Continuous Assessment Tool (CAT)**: It may include: Quiz, Project based learning, Case study, Seminar, or Open Book Assignment

   c. One **AI-powered Assignment (APA):** Use of AI tools for Mini-project to analyze code efficiency, estimate energy consumption and Identify inefficient components. Apply improvements using green coding practices and Compare **before vs after** results.

2. **"External Assessment"** refers to the Semester End Examinations (SEE). The weightage of this component is **40%**. At the end of semester, the final examination is conducted to evaluate

student's performance.

A. Professional Core Courses (Theory)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT)	20
Mid Semester Examination (MSE)	30
AI-powered Assignment "Or" Mini Project Ø Energy Monitoring & Data Analysis Ø AI-Based Optimization Engine	10
Total	**60 marks**

CO	Outcome	Bloom's Taxonomy Level
CO1	Examine the concepts of Green IT, environmental impacts of computing, and the role of IT in achieving sustainable development.	L4
CO2	Apply principles of green software engineering to design energy-efficient and environmentally sustainable software solutions.	L3
CO3	Analyze and optimize software code using energy-efficient coding practices and carbon measurement techniques.	L4
CO4	Evaluate and implement sustainable practices across the software development lifecycle, including monitoring and optimization.	L5
CO5	Assess the environmental impact of AI/ML systems and apply techniques to develop energy-efficient and sustainable intelligent solutions.	L6

Strength of CO Mapping to PO/PSOs:

Table: Mapping Levels of Cos to Pos / PSOs

COs \ POs/PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	1	–	1	2	3	2	–	1	–	1	1	–
CO2	2	2	3	1	2	1	3	2	1	1	1	1	2	2
CO3	2	3	2	2	3	–	2	1	–	1	–	1	3	2
CO4	2	2	3	2	2	2	3	2	1	1	2	1	2	2
CO5	2	3	2	2	3	2	3	3	1	1	1	2	3	3
Avg	10/15	12/15	11/15	7/15	11/15	7/15	14/15	10/15	3/15	5/15	4/15	6/15	11/15	10/15

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Recommended Tools

- AI Tools: CodeCarbon, Green Metrics Tool, Eco2AI, Green Algorithms, Persefoni Pro, Nvidia CUDA tool kit.

Text Books

1. San Murugesan and G.R. Gangadharan, "Harnessing Green IT: Principles and Practices", IEEE, Wiley, 2013.
2. Anne Currie, Sarah Hsu & Sara Bergman , "Building Green Software: A Sustainable Approach to Software Development and Operations", O'Reilly, 2024, Working Program Ltd.,

The Writer's House LTD, and Sara Bergman AS

Additional Resources

1. Jason Harris - Green Computing and Green IT Best Practices on Regulations and Industry Initiatives, Virtualization, Power Management, Materials Recycling and Telecommuting (2008, Emereo Publishing) - Libgen.lc.
2. Robert C. Martin, "Clean Code: A Handbook of Agile Software Craftsmanship", Stanford, O'Reilly, 2021, 2^{nd} edition.
3. Roger S. Pressman, "Software Engineering: A Practitioner ' s Approach", McGraw-Hill, 7^{th} edition .

****

Lean Start-up Methodology and Product Marketing

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L+15 Hrs P
L–T–P-J	:	3–0–0-2			

Course Vision:

The course aims to develop entrepreneurial thinking, innovation, and product marketing skills required to build and scale customer-centric business ventures using lean startup methodologies. It enables students to design and validate business models, apply storytelling and pitching techniques, and achieve product-market fit through experimentation and customer feedback. The course also equips learners with modern tools, AI-driven decision-making approaches, and strategic communication skills necessary for developing sustainable and innovative startup solutions.

Course Learning Objectives:

This Course will enable students to:

1. Understand and apply lean startup principles to develop innovative business ideas.
2. Analyze and design business models using structured frameworks.
3. Develop and validate solutions through experimentation and customer feedback.
4. Apply storytelling techniques to effectively pitch startup ideas.
5. Understand product marketing strategies and evaluate product-market fit.
6. Use modern tools and AI to enhanceinnovation, decision-making, and communication.

Broad Course Description

This course introduces students to the principles of lean startup thinking, focusing on iterative development, validated learning, and customer-centric innovation. It integrates startup methodologies with product marketing strategies to help learners design, test, and scale business ideas effectively. Students will explore business model innovation, storytelling for pitching, and techniques for achieving product-market fit while using modern tools and data-driven decision-making.

UNIT – I	07 Hours

VISION:

Start, Define, Learn & Experiment.

UNIT – II	09 Hours

STEER:

Leap, Test, Measure & Pivot

ACCELERATE:

Batch, Grow, Adapt, Innovate

UNIT – III	09 Hours

BUSINESS MODEL PART I

Business Model Definition, 9 Building Blocks, Business Model Canvas, Patterns

UNIT – IV	10 Hours

BUSINESS MODEL PART II

Design – Customer insights, Ideation, Prototyping, Story Telling
Strategy – BM environment, Evaluating BM’s, Manage
Process – Business Model Design Process

UNIT – V	10 Hours

Introduction to Product Marketing, Visual representation of Product Marketing, building a Customer Persona, Product-Market Fit, Product Life Cycle, Product Marketing vs Regular Marketing, Responsibilities of a Product Marketer, Critical steps of Product Marketing, Measuring the impact of Product Marketing.

Resources:

Video: https://olympus.mygreatlearning.com/courses/91026/pages/introduction-to-product-marketing?module_item_id=3776737

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Enabled Startup Ideation & Validation:**

Leverage AI tools to generate, refine, and validate startup ideas using market signals, customer insights, and trend analysis while ensuring originality and contextual relevance.

- **Data-Driven Decision Making in Lean Cycles:**

Apply AI for analyzing customer feedback, experiments, and performance metrics to support the build–measure–learn cycle and make informed pivot or persevere decisions.

- **AI-Assisted Business Model Innovation:**

Utilize AI to explore and evaluate business model alternatives, optimize value propositions, and simulate different market scenarios for strategic decision-making.

- **Intelligent Customer Persona & Market Analysis:**

Use AI to develop detailed customer personas, segment markets, and predict user behavior, while critically assessing the accuracy and bias in generated insights.

- **AI-Enhanced Storytelling & Pitch Design:**

Employ AI tools to structure compelling startup pitches, refine narratives, and improve communication effectiveness, ensuring authenticity and personal interpretation.

- **Responsible & Ethical Use of AI in Entrepreneurship:**

Ensure ethical use of AI by validating outputs, avoiding over-reliance, maintaining transparency, and reflecting on biases, limitations, and the role of human judgment in decision-making.

Suggested Tools:

- Miro
- Notion
- FigJam
- Strategyzer
- Figma
- Survey & Validation: Typeform

Applications:

- Startup development and entrepreneurship
- Product management roles
- Digital marketing strategy
- Innovation and consulting domains
- Corporate intrapreneurship

Project Component

- Develop a startup idea and validate it using lean startup methodology.
- Create a complete Business Model Canvas for a new or existing startup.
- Design a product marketing strategy for a newly launched product.
- Conduct customer discovery and build a validated customer persona.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Explain lean startup concepts such as build-measure-learn and pivot strategies	L2
CO2	Develop and evaluate business models using the Business Model Canvas	L3
CO3	Apply customer discovery and validation techniques to test ideas	L4
CO4	Design and deliver effective startup pitches using storytelling	L5
CO5	Analyze product marketing strategies and measure product performance	L5

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**										**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			3			2	2	2	2		3
CO2	3	3	2		3			2	2	2	2		3
CO3	2	3	3	2	3			2	2	2	2		3
CO4	2	2	3	3	3			2	2	2	2		3
CO5	3	3	2	2	3			2	2	2	2		3

3: Substantial (High)   2: Moderate (Medium)   1: Poor (Low)

TEXT BOOKS:

1. Eric Ries "The Lean Startup", ISBN–978-0670921607. https://ia801206.us.archive.org/31/items/TheLeanStartupErickRies/The%20Lean%20Startup%20-%20Erick%20Ries.pdf

2. Alex Osterwalder, Yves Pigneur "Business Model Generation Handbook" ISBN-978-0-470-87641-1 http://alvarestech.com/temp/PDP2011/pdf/Business%20Model%20Generation%20(1).pdf

REFERENCE BOOKS:

1. Philip Kotler, Kevin Lane Keller, Abraham Koshy and Mithileswar Jha, Marketing Management, Pearson, 14th Edition, 2013.
2. Steve Blank "The Four Steps to the Epiphany: Successful Strategies for Products That Win", ISBN-13 : 978-0989200509
3. The Entrepreneur's Guide to Customer Development: A "Cheat Sheet" to the Four Steps to the Epiphany Paperback – Import, 6 February 2012, **ISBN-13 :** 978-

0982743607

E-RESOURCES:

1. https://learning.productmarketingalliance.com/on-demand/advanced-product-marketing- certified.
2. https://learning.productmarketingalliance.com/leadership-on-demand/product-marketing- certified-leadership#course-overview.
3. https://ia801206.us.archive.org/31/items/TheLeanStartupErickRies/The%20Lean%20 Startup%20-%20Erick%20Ries.pdf
4. http://alvarestech.com/temp/PDP2011/pdf/Business%20Model%20Generation%20()

   .pdf
5. https://www.getstoryshots.com/books/the-lean-startup-summary/
6. https://www.strategyzer.com/canvas

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving with on field exposure
2. Build Start-up and test the solution with real customers.
3. Real-world case studies of successful and unsuccessful product launches.
4. Brainstorm different messaging strategies for various customer segments

****

Natural Language Processing for GenAI

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L +15Hrs P
L–T–P-J	:	2–0–2-0			

Course Vision: The course aims to develop strong foundations in Natural Language Processing and Generative AI for building intelligent language-based systems using modern AI techniques. It enables students to understand and implement neural language models, Transformer architectures, and Large Language Models for tasks such as text generation, summarization, question answering, and code generation. The course also equips learners with practical skills in prompt engineering, fine-tuning, and responsible AI practices required for developing ethical, scalable, and real-world AI-driven NLP applications.

Prerequisites: Basic programming knowledge, Machine Learning, Probability and Statistics.

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamental concepts of Natural Language Processing, language models, and the evolution of Generative AI.
2. To analyze word embeddings, neural language models, and Transformer-based architectures for NLP tasks.
3. To design and implement solutions using Large Language Models (LLMs) such as GPT and BERT with appropriate tokenization techniques.
4. To evaluate prompt engineering strategies and advanced architectures including encoder–decoder models and fine-tuning approaches.
5. To apply Generative AI techniques for building intelligent NLP applications such as summarization, question answering, and code generation.

Broad Course Description

This course provides a comprehensive foundation in Generative AI for Natural Language Processing (NLP), covering the evolution from traditional NLP techniques to modern deep learning and Transformer-based architectures. Students will explore word embeddings, neural language models, and Large Language Models (LLMs), along with key concepts such as prompt engineering, fine-tuning, and multimodal generation. The course emphasizes practical implementation of AI-driven NLP applications including text summarization, question answering, and code generation. It also addresses ethical considerations, biases, and responsible AI practices, while fostering hands-on learning through industry-relevant projects and AI-assisted pedagogy.

UNIT – I	06 Hours

Traditional NLP – Foundations:

Fundamentals of traditional Natural Language Processing (NLP) including its scope, history, and applications, text pre-processing techniques such as tokenization, stemming, lemmatization, and stop-word removal, Part-of-Speech (POS) tagging using rule-based and statistical methods, Named Entity Recognition (NER) with IOB tagging and Conditional Random Fields (CRF).

[T1] Ch. 1–3, 6 Speech and Language Processing – Jurafsky & Martin
[T2] Ch. 1–4 Foundations of Statistical NLP – Manning & Schütze
[R1] Ch. 1 Natural Language Processing with Python – Bird et al. (NLTK Book)

UNIT – II	07 Hours

Traditional NLP – Information Extraction & Classical ML:

Syntactic analysis using Context-Free Grammar (CFG), CYK algorithm, and dependency parsing, lexical semantics using WordNet and ontologies, N-gram language models with Maximum Likelihood Estimation (MLE), smoothing techniques (Laplace, Kneser-Ney), perplexity, and document representation using Bag-

of-Words and TF-IDF. Sentiment analysis using lexicon-based and machine learning-based approaches such as VADER, feature engineering for text including n-gram, character, and linguistic features, information extraction tasks such as relation extraction, event extraction, and coreference resolution, fundamentals of machine translation including rule-based methods, word alignment, and IBM models, question answering systems based on passage retrieval and pattern matching, and evaluation metrics such as accuracy, precision, recall, F1-score, BLEU, and ROUGE.
[T1] Ch. 4–5, 17, 23 Speech and Language Processing – Jurafsky & Martin
[T2] Ch. 5–7, 12 Foundations of Statistical NLP – Manning & Schütze [R2]
Ch. 1–3 NLP with PyTorch – Rao & McMahan

UNIT – III	07 Hours

Modern NLP – Deep Learning & Transformers:
Neural and advanced NLP techniques including word embeddings such as Word2Vec (CBOW, Skip-gram), GloVe, and FastText, recurrent architectures including Vanilla RNN, LSTM, and GRU for sequence labelling and classification, sequence-to-sequence models and encoder–decoder architecture, attention mechanisms including Bahdanau attention and self-attention, Transformer architecture with multi-head attention, positional encoding, and feed-forward layers, pre-trained language models such as BERT, GPT, RoBERTa, and XLNet with pre-training objectives, transfer learning and fine-tuning on NLP benchmarks like GLUE and SQuAD, and advanced NLP tasks including text summarization (extractive and abstractive) and dialogue systems.
[T1] Ch. 7–11, 26 Speech and Language Processing – Jurafsky & Martin
[T3] Ch. 10–12, 16 Deep Learning – Goodfellow, Bengio & Courville [R2]
Ch. 4–8 NLP with PyTorch – Rao & McMahan
[R3] Ch. 1–4 Natural Language Processing with Transformers – Tunstall et al.

UNIT – IV	05 Hours

Overview of AI and Generative AI:
Introduction to AI, Machine Learning (ML), and Deep Learning (DL), Supervised, Unsupervised and Reinforcement learning , Neural network as AI model, Why Learn Generative AI? , Introduction to Generative AI and its Capabilities , What Makes Generative AI Different from Traditional AI? , Evolution of Generative AI.
(Text Book-3: Chapter 20: 20.1, 20.2, 20.3)
(Text Book-4: Chapter 1: 1.1, 1.2, 1.3)
(Reference Book-4: Chapter 1: 1.1, 1.2)
(Reference Book-5: Chapter 1: 1.1)

UNIT – V	05 Hours

Tools and Models in Generative AI:
Overview of Generative Models (GANs, VAEs), Introduction to Large Language Models (LLMs), Tools for Generating Text, Images, Audio, Video, and Code, Basics of Prompt Engineering, Applications of Generative AI.
(Text Book-3: Chapter 20: 20.4, 20.5)
(Text Book-4: Chapter 2: 2.1, 2.2, 2.3)
(Reference Book-3: Chapter 1: 1.1, 1.2)
(Reference Book-5: Chapter 2: 2.1, 2.2)

DLI: Introduction to Generative AI, Word Embeddings, Tokens, and NLP and Large Language

Lab Experiments:

1. **Text Preprocessing Pipeline Implementation**
2. Perform tokenization, stop-word removal, stemming, and lemmatization on a sample corpus using NLTK.

3. **Part-of-Speech (POS) Tagging using Rule-based and Statistical Methods**
4. Implement and compare POS tagging techniques on a given dataset.
5. **Named Entity Recognition (NER) using CRF**
6. Build an NER model with IOB tagging using Conditional Random Fields.
7. 4.**N-gram Language Model with Smoothing Techniques**
8. Implement unigram, bigram models with Laplace smoothing and compute perplexity.
9. **Document Representation using Bag-of-Words and TF-IDF**
10. Convert text data into feature vectors and analyze differences.
11. **Text Classification using Naïve Bayes and Logistic Regression**
12. Build and evaluate classifiers on a dataset like spam detection or sentiment analysis.
13. **Sentiment Analysis using Lexicon-based (VADER) and ML-based Approaches**
14. Compare performance of rule-based and machine learning models.
15. **Information Extraction: Relation or Event Extraction**
16. Implement a simple pipeline for extracting relationships or events from text.
17. **Transformer-based Text Classification using Pre-trained Models (e.g., BERT)**
18. Fine-tune a transformer model for classification tasks.
19. **Generative AI Application using Prompt Engineering**
20. Design prompts for tasks like text generation, summarization, or code generation using LLM tools.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy):

- AI-assisted understanding of traditional NLP techniques including text pre-processing, POS tagging, NER, parsing, and language models.
- AI-driven implementation of NLP applications such as text classification, sentiment analysis, information extraction, and machine translation.
- AI-enabled exploration of word embeddings, neural language models, and Transformer-based architectures for advanced NLP tasks.
- AI-guided understanding of Generative AI concepts, including GANs, VAEs, and the evolution of generative models.
- AI-based application of Large Language Models (LLMs), prompt engineering, and multimodal generative tools for text, image, audio, and code generation.

- Students evaluate challenges in traditional NLP such as ambiguity, sparsity, and limitations of rule-based and statistical methods.

- Students analyze performance, biases, and limitations of classical NLP applications and evaluation metrics.
- Students assess efficiency, scalability, and limitations of embeddings, neural models, and Transformer architectures.
- Students examine ethical concerns, misuse risks, and societal impact of Generative AI technologies.
- Students evaluate reliability, interpretability, and ethical implications of LLMs, prompt engineering, and AI-generated content.

Applications:

- Language Modeling Systems (Word Embeddings, Neural Language Models, NLP-based Applications)
- Large Language Models (GPT, BERT, Tokenization Techniques, Small Language Models)

- Advanced NLP Architectures (Encoder–Decoder Models, BART, T5, Parameter-Efficient Fine-Tuning, Model Optimization)
- Prompt Engineering Systems (Prompt Design, Control Strategies, Task-Specific Prompting)
- AI-driven NLP Applications (Summarization, Question Answering, Code Generation, Reasoning, Advanced Prompting Frameworks like CoT, PAL, ReAct)

Mini Project Component

Some sample projects described below simulate the real software engineering problems.

Project 1 – Text Classification & Sentiment Analysis System.

- Problem: Develop a system to classify text documents and analyze sentiment from user-generated content.
- Concepts used: Naïve Bayes, Logistic Regression, SVM, feature engineering (n-grams, TF-IDF), sentiment analysis techniques (lexicon-based, ML-based).
- Industry relevance: Social media analytics, customer feedback systems, companies like Amazon, Twitter.

Project 2 – Named Entity Recognition & Information Extraction System

- Problem: Build a system to extract entities, relationships, and events from unstructured text data.
- Concepts used: POS tagging, NER (IOB tagging, CRF), parsing, relation extraction, coreference resolution.
- Industry relevance: Healthcare data processing, legal document analysis, enterprise knowledge extraction systems.

Project 3 – Machine Translation & Question Answering System

- Problem: Develop a system capable of translating text between languages and answering questions based on given context.
- Concepts used: Rule-based MT, word alignment, basic translation models, passage retrieval, pattern matching, evaluation metrics (BLEU, ROUGE).
- Industry relevance: Multilingual platforms, search engines, virtual assistants (Google Translate, chatbots).

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain fundamentals of Natural Language Processing (NLP), including text pre-processing, linguistic analysis, and classical language models.	L2
CO2	Apply machine learning techniques for NLP tasks such as text classification, sentiment analysis, and information extraction.	L3
CO3	Design and utilize advanced NLP models including word embeddings, neural language models, and Transformer architectures.	L3
CO4	Analyze and implement Generative AI concepts including GANs, VAEs, and foundational generative models.	L3
	Develop AI-driven NLP applications using Large Language Models	

CO5	(LLMs), prompt engineering, and multimodal generative techniques.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	2				2		2	2	2
CO2	3	2	2	2	2				2		2	2	2
CO3	3	2	2	2	2				2		2	2	2
CO4	3	2	2	2	2				2		2	2	2
CO5	3	2	2	3	2				2		2	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Speech and Language Processing – Daniel Jurafsky and James H. Martin, 3rd Edition (Draft),PrenticeHall/Online.
2. Comprehensive resource covering fundamentals of NLP, language models, and modern deep learning approaches.
3. Foundations of Statistical Natural Language Processing – Christopher D. Manning and Hinrich Schütze, MIT Press, Cambridge, Massachusetts & London, England. Focuses on Transformer models, BERT, GPT, and real-world NLP applications.
4. Deep Learning – Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016
5. Generative Deep Learning – David Foster, 2nd Edition, O’Reilly Media, April 2023.

REFERENCE BOOKS:

6. Natural Language Processing with Python (NLTK Book) – Steven Bird, Ewan Klein and Edward Loper, Online Edition, O’Reilly / nltk.org.
7. Natural Language Processing with PyTorch – Delip Rao and Brian McMahan, 1st Edition, O’Reilly Media.
8. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra and Thomas Wolf, Revised Edition, O’Reilly Media.
9. Designing Large Language Models – Sinan Ozdemir, 1st Edition, O’Reilly Media.
10. AI Engineering: Building Applications with Foundation Models – Chip Huyen, 1st Edition, O’Reilly Media.

E-RESOURCES:

1. NPTEL (Courses on NLP, Machine Learning, Deep Learning, and Generative AI by IITs)
2. MIT OpenCourseWare (Courses on Machine Learning, Natural Language Processing, and Artificial Intelligence fundamentals)
3. IEEE Xplore (Research articles on NLP, Large Language Models, and Generative AI)

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion. E.g., Designing NLP-based

applications such as text classification, sentiment analysis, or information extraction systems.

2. Role play. E.g., Students act as tokenizer, POS tagger, parser, and classifier to simulate the flow of a traditional NLP pipeline.
3. Demonstration of solution to a problem through programming. E.g., Implementing NLP models using Python (NLTK / Scikit-learn) for tasks like classification, sentiment analysis, or NER.
4. Flip class activity. E.g., Feature engineering techniques, machine learning models for NLP, evaluation metrics, and real-world NLP applications.

****

Theory of computation and Compiler Design

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15Hrs T
L–T–P-J	:	2–1–0-0			

Course Vision:

The course aims to develop strong theoretical and practical foundations in computation models, formal languages, and compiler design for understanding the principles of programming language processing and computation. It enables students to analyze automata, grammars, computability, and decidability concepts while applying compiler construction techniques such as lexical analysis, parsing, optimization, and code generation. The course also equips learners with analytical and problem-solving skills required for designing efficient language processors, compilers, and advanced computing systems.

Prerequisites: Discrete Mathematics, Data Structures

Course Learning Objectives:

This Course will enable students to:

1. Understand formal models of computation
2. Analyze different classes of languages and automata
3. Apply concepts of computability and decidability
4. Understand phases of compiler design
5. Design lexical analyzers and parsers
6. Apply optimization and code generation techniques

Broad Course Description

The course Theory of Computation and Compiler Design provides a comprehensive understanding of formal computational models and their application in compiler construction. It covers automata theory including finite automata, context-free grammars, and Turing machines to analyze computational problems and language recognition. The course also introduces compiler design principles, including lexical analysis, syntax analysis, intermediate code generation, optimization, and target code generation.

UNIT – I	10 Hours

Finite Automata and Regular Languages:

Introduction to Theory of Computation, alphabets, strings, and languages, deterministic finite automata (DFA), non-deterministic finite automata (NFA), ε-NFA, equivalence of DFA and NFA, regular expressions, conversion between regular expressions and finite automata, closure properties, pumping lemma, applications.

(Text Book-1: Chapter 1, Chapter 2, Chapter 3)

UNIT – II	09 Hours

Context-Free Grammars and Pushdown Automata:

Context-free grammars (CFG), derivations, parse trees, ambiguity, simplification of CFG, elimination of ε-productions and unit productions, normal forms (CNF, GNF), pushdown automata (PDA), acceptance criteria, equivalence of CFG and PDA, applications

(Text Book-1: Chapter 4, Chapter 5)

UNIT – III	08 Hours

Turing Machines and Computability:

Turing machines, design and representation, variants of TM, Church-Turing thesis, decidability, halting problem, Post correspondence problem, recursive and recursively enumerable languages, introduction

to complexity (P, NP). (Text Book-1: Chapter 6, Chapter 7, Chapter 8)	
UNIT – IV	**10 Hours**
Compiler Design – Lexical and Syntax Analysis: Introduction to compilers, phases of compiler, lexical analysis (tokens, patterns, lexemes, finite automata, Lex tool), syntax analysis (CFG, parsing techniques – LL, LR), parse trees, syntax trees, YACC basics. (Text Book-2: Chapter 1, Chapter 3, Chapter 4)	
UNIT – V	**08 Hours**
Intermediate Code Generation, Optimization and Code Generation: Intermediate code generation (three-address code, syntax-directed definitions), code optimization (basic blocks, flow graphs, loop optimization, dead code elimination), code generation (register allocation, instruction selection), runtime environment. (Text Book-2: Chapter 6, Chapter 8, Chapter 9)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Automata Design**: Use AI tools to generate and validate automata and grammars.
- **Compiler Automation using AI**: Apply AI tools for generating lexical analyzers and parsers.
- **Simulation-Based Learning**: Use simulation tools to visualize automata and Turing machines.
- **AI-Augmented Code Optimization**: Utilize AI for improving intermediate code and optimization strategies.
- **Hybrid AI–Compiler Systems**: Integrate AI techniques with compiler design for intelligent code transformation.
- **Responsible Computing**: Ensure ethical considerations in compiler optimization and automated systems.

Suggested Tools:

- GitHub Copilot
- JFLAP
- Automata Tutor
- Lex
- Yaac

- GeeksForGeeks

Applications:

1. Compiler construction and programming language design
2. Pattern recognition and text processing
3. Syntax analysis in programming languages
4. Artificial Intelligence and Natural Language Processing
5. Software optimization and code generation
6. Formal verification and system design

Mini Project Component

Project 1 – Automata Simulator

- **Problem:** Design a simulator for DFA/NFA/PDA
- **Concepts used:** Automata, language recognition
- **Industry relevance:** Pattern matching, compilers

Project 2 – Mini Compiler Design

- **Problem**: Develop a simple compiler (lexical + syntax analyzer)
- **Concepts used**: Lex, YACC, parsing
- **Industry relevance**: Programming languages, software tools

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain automata and formal languages	L2
2	Apply grammars and automata models	L3
3	Analyze computational problems and machines	L4
4	Evaluate compiler design techniques	L5
5	Design compiler components and optimization techniques	L6

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2			2							2	2
CO2	3	3	2	2	3							3	2
CO3	2	3	2	2	3							3	2
CO4		2			2	3						2	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Michael Sipser, *Introduction to the Theory of Computation*, 3rd ed., Boston, MA, USA: Cengage Learning, 2012.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd ed., Boston, MA, USA: Pearson Education, 2006.

REFERENCE BOOKS:

1. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed., Boston, MA, USA: Pearson, 2008.
2. Kenneth C. Louden, *Compiler Construction: Principles and Practice*, Boston, MA, USA: Cengage Learning, 1997.
3. K. L. P. Mishra and N. Chandrasekaran, *Theory of Computer Science: Automata, Languages and Computation*, 3rd ed., New Delhi, India: PHI Learning, 2012.
4. J. D. Ullman, *Elements of Compiler Design*, New Delhi, India: Narosa Publishing House, 2007.
5. E. Balagurusamy, *Programming in ANSI C*, 7th ed., New Delhi, India: McGraw Hill Education, 2017. *(for compiler programming background)*

E-RESOURCES:

- https://nptel.ac.in
- https://www.geeksforgeeks.org
- https://ocw.mit.edu
- https://www.javatpoint.com/compiler-design

Activity Based Learning (Suggested Activities in Class)

Activity 1: Automata design and simulation
Activity 2: Grammar construction and parsing
exercise Activity 3: Lex and YACC implementation
Activity 4: Mini compiler development

****

Computer Vision

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P–J	:	3–0–0–0			

Course Vision: The course aims to develop strong foundations in deep learning and computer vision for designing intelligent systems capable of understanding and analyzing visual data. It enables students to build and apply neural network models such as CNNs, RNNs, and LSTMs for image classification, object detection, segmentation, and sequential vision tasks. The course also equips learners with practical and research-oriented skills required for developing AI-driven vision solutions in domains such as healthcare, autonomous systems, surveillance, and multimedia analytics.

Prerequisites: Machine Learning, Deep Learning

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamentals of neural networks and deep learning
2. To develop CNN-based models for image classification, detection, and segmentation
3. To implement sequential models (RNN/LSTM) for vision tasks
4. To apply deep learning techniques to real-world datasets
5. To promote research and innovation in AI-driven vision systems.

Broad Course Description

This course introduces the core concepts of deep learning with a focus on computer vision. It covers neural network training, convolutional architectures, object detection, image segmentation, and sequence modelling using RNNs. Students gain hands-on experience in building, training, and evaluating models for real-world applications such as autonomous systems, healthcare imaging, and video analytics.

UNIT – I	09 Hours

MODULE 1: FOUNDATIONS OF NEURAL NETWORKS

Introduction, Feedforward Neural Networks, Activation Functions and Cost Functions, Backpropagation, Stochastic Gradient Descent and its Variants, Need for Regularization, Regularization Techniques (L1, L2, Dropout), Improving Training of Neural Networks, Code Walkthrough: Training and Testing Pipeline with Simple Dataset.

(Text Book-1,2 and 3)

UNIT – II	09 Hours

MODULE 2: CNNS FOR IMAGE CLASSIFICATION

Introduction to CNNs, Convolution and Pooling Layers, Backpropagation in CNNs, Data Augmentation Techniques, Transfer Learning and Fine-tuning, Visualization Techniques, Case Studies: VGGNet, Inception-V3, ResNet-50, MobileNet

(Text Book-1,2 and 3)

UNIT – III	08 Hours

MODULE 3: CNNS FOR OBJECT DETECTION

Evaluation Metrics: IoU and mAP, Localization Techniques: Sliding Window and Regression Approach, Single-stage Detectors: YOLO, Two-stage Detectors: Fast R-CNN and Faster R-CNN, Single Shot

MultiBox Detector (SSD). (Text Book-1,2 and 3)	
UNIT – IV	**09 Hours**
MODULE 4: CNNS FOR IMAGE SEGMENTATION Segmentation Concepts, Evaluation Metrics for Segmentation, Fully Convolutional Networks (FCN), SegNet Architecture, Dilated Convolutions, Case Studies: U-Net, Mask R-CNN. (Text Book-1,2 and 3)	
UNIT – V	**10 Hours**
MODULE 5: RECURRENT NEURAL NETWORKS FOR VISION Introduction to RNNs, Backpropagation Through Time (BPTT), Long Short-Term Memory (LSTM), Temporal Modeling in Vision, Case Study: Video Understanding using CNNs and RNNs. (Text Book-1,2 and 3)	
DLI: Applied Image Classification. Applied Object Detection. Convolutional Neural Networks. and Recurrent Neural Networks.	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy):

- Optimization algorithms, regularization, model training pipelines
- Transfer learning, feature extraction, pretrained models
- YOLO, Faster R-CNN, SSD models
- Pixel-level classification, U-Net, Mask R-CNN
- LSTM, sequence modeling, temporal learning
- Human learning vs gradient-based learning.
- Visual perception vs learned feature hierarchies
- Attention and localization in human vision vs AI.
- Scene understanding vs pixel-wise reasoning
- Memory and sequence learning in human's vs AI.

Applications:

8) Signal classification, predictive analytics, pattern recognition
9) Face recognition, defect detection, medical image classification

10) Autonomous vehicles, surveillance, robotics vision

11) Medical imaging, satellite image analysis, scene understanding
12) Video analytics, action recognition, gesture recognition

Mini Project Component:

)Real-time object detection system using webcam

a) Medical image segmentation prototype

b) Video-based action recognition system using CNN + LSTMIntelligent safety/security system.

c) Implement object detection using YOLO or SSD on a custom dataset

d) Perform image segmentation using U-Net and analyze results

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain and implement neural network training mechanisms	L3
CO2	Design CNN models for image classification tasks	L6
CO3	Develop object detection models using modern frameworks	L3
CO4	Apply image segmentation techniques	L4
CO5	Apply RNN/LSTM models for video and sequential vision tasks	L4

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	3	–	–	–	2	2	–	2	3
CO2	3	3	3	2	3	2	–	–	2	2	–	2	3
CO3	3	3	3	3	3	2	2	2	2	2	2	2	3
CO4	3	3	3	3	3	2	2	2	2	2	2	2	3
CO5	2	3	3	3	3	2	2	2	3	3	2	2	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Textbooks:

1. Deep learning for computer vision, R. Shanmugamani, Packt Publishing, 2018.
2. Deep Learning, I. Goodfellow, Y. Bengio, A. Courville, MIT Press, 2016.

REFERENCE BOOKS:

1. Materials from Hugging Face: https://huggingface.co/learn/computer-vision course/

E resources:

1. IEEE research articles
2. NPTEL course on Deep Learning for Computer Vision by Dr. Vineet Balasubramanian: https://onlinecourses.nptel.ac.in/noc24_cs89/preview

Activity Based Learning (Suggested Activities in Class)

5. Implement and Compare Activation Functions in a Neural Network
6. Build and Train a CNN Model for Image Classification using a Real Dataset
7. Object Detection using Pre-trained YOLO Model on Custom Images
8. Image Segmentation using U-Net Architecture on Medical/Real-world Data
9. Video Classification using CNN + LSTM for Temporal Feature Learning.

****

Cryptography & Cyber Security

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15 HrsJ
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop strong foundations in modern cryptography and secure communication systems for protecting data confidentiality, integrity, and authenticity in digital environments. It enables students to understand and implement cryptographic algorithms, digital signatures, PKI, and secure communication protocols for enterprise and IoT applications. The course also prepares learners to address emerging cybersecurity challenges through post-quantum cryptography, hardware-accelerated security, and AI-integrated cryptographic solutions.

Course Learning Objectives:

This Course will enable students to:

1. Master principles of symmetric and asymmetric cryptography.
2. Implement digital signatures, message authentication, and hash algorithms.
3. Understand key management, PKI, and internet security protocols.
4. Explore post-quantum, hardware-accelerated, and AI-linked cryptography.
5. Apply cryptographic design for secure enterprise and IoT systems.

Broad Course Description

This course explores mathematical foundations and practical implementations of cryptographic systems that ensure data confidentiality, integrity, and authenticity. Students learn symmetric and asymmetric algorithms, digital signatures, PKI, and secure communication protocols, linking modern cryptography to enterprise and IoT security requirements in the evolving cyber landscape.

UNIT – I	06 Hours

FOUNDATIONS OF CRYPTOGRAPHY:

Classical ciphers: substitution, transposition, frequency analysis. Modern cryptographic primitives: confidentiality, integrity, authenticity. Symmetric key concepts and random number generation. Modular arithmetic, Euler's theorem, modular inverses, GCD algorithms. Project Activity: Implement Caesar and Vigenère cipher in C/Python with brute-force attack.

(Text Book 1 : Ch. 1 1.1–1.5; Text Book 2 : Ch. 2 2.1–2.3)

UNIT – II	06 Hours

SYMMETRIC CRYPTOGRAPHY:

Feistel network and data encryption standards (DES, 3DES, AES).

Block vs stream ciphers, chaining modes (CBC, CFB, OFB, CTR). Key distribution and session key protocols. Differential and linear cryptanalysis introduction.

(Text Book 1 : Ch. 3 3.1–3.5)

(Text Book 2 : Ch. 3 3.2–3.5)

Mini Lab: Performance comparison of AES and ChaCha20 (crypto libraries benchmarking).

UNIT – III	06 Hours

PUBLIC KEY CRYPTOGRAPHY & HASH FUNCTIONS:

RSA mathematical concept, key generation steps.

Diffie–Hellman key exchange and Elliptic Curve Cryptography (ECC). Message Authentication Codes and Hash functions (MD5, SHA-2/3). Digital signatures and PKI architectures.

(Text Book 1 : Ch. 6 6.1–6.5)

(Text Book 2 : Ch. 5 5.1–5.4)
Activity: Create digital certificate chain using OpenSSL.

UNIT – IV	**06 Hours**
SECURITY PROTOCOLS: Secure Email (PGP, S/MIME), IPSec architecture and components. SSL/TLS handshake and cipher-suite negotiation. Kerberos authentication system. Cryptanalysis labs (using Hashcat). (Text Book 1 : Ch. 8 8.1–8.3) (Text Book 2 : Ch. 7 7.1–7.5) **Project Task:** TLS handshake trace and analysis using Wireshark.	
UNIT – V	**06 Hours**
ADVANCED CRYPTOGRAPHY: Quantum-resistant algorithms (Lattice, Code, Hash based). Blockchain cryptography (review of next domain course links). Hardware crypto modules (HSM, TPM). AI-based key management and anomaly detection. (Text Book 1 : Ch. 10 10.1–10.3) (Text Book 2 : Ch. 9 9.1–9.4) **Capstone:** Implement secure file sharing with AES + RSA hybrid encryption.	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

1. GPU-accelerated encryption (NVIDIA CUDA Crypto).
2. Machine learning for encryption pattern monitoring.
3. Responsibility – privacy-preserving ML.

Applications:

1. Encrypting sensitive communications in banking and e-commerce systems.
2. Enabling secure web transactions through SSL/TLS protocols.
3. Implementing digital signatures and certificates in public key infrastructure.
4. Protecting IoT devices with lightweight cryptographic algorithms.
5. Designing post-quantum and AI-enhanced encryption models.

Mini Project Component

1. VPN prototype using AES + RSA.
2. Password-less authentication through ECC.
3. Timing attack simulation and mitigation.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain crypto principles and mathematical foundations.	L2
CO2	Implement symmetric & asymmetric algorithms practically.	L3
CO3	Apply digital signatures and PKI for security.	L3
CO4	Analyze protocols and demonstrate secure implementations.	L3

CO5	Design hybrid and AI-linked crypto solutions.	L4

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2									2	2	
CO2	2	3	2	2	3						2	3	
CO3	2	2	3	2	3						2	3	
CO4	2	3		3	2						2	2	
CO5	2	2	3		3							3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. William Stallings – Cryptography and Network Security, 8e.
2. Behrouz A. Forouzan – Cryptography and Network Security, McGraw-Hill.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
CUDA Accelerated Crypto and Privacy	II, V	Hardware crypto optimization
Morpheus Cybersecurity Framework	III, IV	AI for malware signature detection

E-RESOURCES:

1. nptel.ac.in
2. coursera.org
3. edx.org
4. cryptohack.org
5. developer.nvidia.com

Activity-Based Learning (Suggested Activities in Class)

1. Implement Caesar, Vigenère, and AES encryption using Python.
2. Demonstrate RSA key generation and message encryption.
3. Capture an HTTPS handshake and decode TLS parameters.
4. Create and verify a digital signature using OpenSSL.
5. Debate on the future of “Quantum-Safe Cryptography.”

****

Gaming Design Fundamentals

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	04 Hours	**Total Hours**	:	30Hrs L+15Hrs J
L–T–P-J	:	2–0-0-2			

Course Vision: The course aims to develop creative, analytical, and technical skills required for designing engaging and immersive game experiences. It enables students to understand game mechanics, storytelling, level design, player psychology, and interactive system design using modern prototyping and design tools. The course also equips learners with the ability to create professional game design documents, develop innovative game concepts, and apply ethical and responsible game design practices for modern digital gaming environments.

Course Learning Objectives:

This Course will enable students to:

1. Understand core principles of game design including the MDA (Mechanics, Dynamics, Aesthetics) framework.
2. Develop skills in game concept development, level design, difficulty tuning, and player experience design.
3. Learn narrative design, character development, branching storytelling, and world-building techniques.
4. Create professional Game Design Documents (GDD) and prototype interactive game concepts.

Broad Course Description

Gaming Design Fundamentals provides a comprehensive foundation in the art and science of designing engaging, playable games. The course approaches game design from both analytical and creative perspectives, examining what makes games compelling and how designers intentionally craft player experiences. Students study the MDA framework, flow theory, Bartle's player taxonomy, game mechanics and economy design, level design principles, narrative design (branching narrative with Twine), world-building, playtesting methodologies, and professional GDD writing. Digital prototyping tools including Figma and GDevelop are used alongside paper prototyping. The course integrates AI tools for concept generation while critically examining responsible and ethical game design practices including addiction mechanics and representation.

UNIT – I	09 Hours

Foundations of Game Design: What is a game – definitions, magic circle, play theory. History of game design from board games to video games. MDA Framework – Mechanics, Dynamics, Aesthetics. Core game loops and feedback loops. Game genres and their design conventions. Player psychology – motivation, challenge, and reward. Flow theory (Csikszentmihalyi) applied to game design. Types of fun – Bartle's player taxonomy (Killers, Achievers, Explorers, Socializers). (Ref: Jesse Schell, The Art of Game Design Ch. 1–5)

Textbook Reference:

• Jesse Schell, The Art of Game Design: A Book of Lenses, 3rd Ed – Ch. 1–5 (CRC Press, 2020)

• Richard Rouse III, Game Design: Theory and Practice, 2nd Ed – Ch. 1–3 (Wordware, 2005)

UNIT – II	09 Hours

Game Mechanics & Systems Design: Game mechanics – rules, goals, actions, resources. Space and time in games. Randomness and probability – dice, cards, procedural generation. Game economy – resources, sinks, sources, converters. Progression systems – leveling, skill trees, unlocks. Balancing game systems – tools and iteration techniques. Emergent gameplay vs. scripted experiences. Core mechanic design exercise through paper prototyping. (Ref: Jesse Schell Ch. 8–14)

Textbook Reference:

• Jesse Schell – Ch. 8–14 • Brenda Romero & Ian Schreiber, Challenges for Game Designers – Ch. 1–5 (Cengage, 2009)	
UNIT – III	**09 Hours**
Level Design & Spatial Design: Principles of level design – guidance, pacing, discovery, and flow. Tutorial design – teaching through play without explicit instruction. Spatial design considerations for 2D, 2.5D, and 3D environments. Environmental storytelling and atmosphere. Landmark design and wayfinding. Difficulty curves and challenge scaling strategies. Reward placement and collectibles design. Level design analysis using case studies from iconic games. (Ref: Scott Rogers, Level Up! Ch. 1–6) Textbook Reference: • Scott Rogers, Level Up! The Guide to Great Video Game Design, 2nd Ed – Ch. 1–6 (Wiley, 2014) • Jesse Schell – Ch. 18–19	
UNIT – IV	**09 Hours**
Narrative Design & World Building: Story vs. Narrative in games – ludonarrative harmony and dissonance. Story structures – three-act structure, Hero's Journey, non-linear narratives. Character design – archetypes, motivation, and backstory. Dialogue writing and branching narrative systems using Twine. World-building – lore, culture, factions, geography, and history. Cutscenes vs. in-engine environmental storytelling. Sound design's role in narrative – ambient, music, and voice acting overview. (Ref: Ernest Adams, Fundamentals of Game Design Ch. 12–14) Textbook Reference: • Ernest Adams, Fundamentals of Game Design, 3rd Ed – Ch. 12–14 (New Riders, 2014) • Twine Documentation (twinery.org)	
UNIT – V	**09 Hours**
Prototyping, Playtesting & GDD: Paper prototyping – rapid ideation and iteration. Digital prototyping tools – Figma, GDevelop, Twine. Playtesting methodologies – observational, think-aloud, feedback surveys. Analyzing playtest data and iterating on designs. Game Design Document (GDD) structure – vision, mechanics, narrative, UI, art direction. Pitching game concepts – elevator pitch and presentation. Ethics in game design – addiction mechanics, representation, microtransactions. Career pathways in game design and industry outlook. (Ref: Jesse Schell Ch. 24–26) Textbook Reference: Ernest Adams – Ch. 4–6 Jesse Schell – Ch. 24–26	

NVIDIA DLI Free Course Mapping (Validated & Justified)

Validation Source: NVIDIA DLI Official Training Catalog (August 2024 PDF) + Live URL verification (April 2026). All courses below are confirmed FREE self-paced courses accessible at **learn.nvidia.com**. Only FREE courses are listed; paid instructor-led workshops excluded.

NVIDIA Deep Learning Institute (DLI) — FREE Self-Paced Course Mapping
Source: NVIDIA DLI Official Catalog (Aug 2024) + Live URL verification (Apr 2026) | Access: learn.nvidia.com | Only FREE self-paced courses included

Unit	NVIDIA DLI Free Course	Category	Duration	Cert.	Verified URL	Mapping Justification
Units	**Generative AI Explained**	Generative AI	1 Hr	—	courses.nvidia.com/course-v1:DLI+S-FX-07+V1	Essential foundation for AI-assisted concept generation, narrative AI, and procedural content creation tools

4–5						evaluated in Units 4–5 — students must understand GenAI before critically using AI game design tools.
Unit 5	**Generative AI With Diffusion Models**	Generative AI / Image	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-v1:DLI+T-FX-11+V1	Diffusion model image generation directly maps to AI-powered concept art generation in Unit 5's prototyping module — students use AI-generated images as concept art within their GDD and Figma prototypes.
Units 4–5	**Prompt Engineering With Llama 2**	Generative AI / LLM	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-	Prompt engineering directly applicable to generating branching game dialogue, quest scripts, character
					v1:DLI+T-FX-12+V1	backstories, and world-lore content for narrative design in Units 4–5.
Unit 5	**Synthetic Data Generation for Training CV Models**	Graphics / CV	2 Hrs	—	learn.nvidia.com/courses/course-detail?course_id= course-v1:DLI+S-OV-24+V1	Synthetic visual data generation using NVIDIA Omniverse — introduces how game-world assets and AI playtesting datasets can be procedurally generated, extending Unit 5's AI playtesting discussion.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Concept Generation:** Use ChatGPT and Midjourney to brainstorm game concepts, character designs, and world lore; critically evaluate and refine AI outputs.
- **Procedural Design Exploration:** Investigate AI-driven procedural content generation and its impact on game replayability, diversity, and designer control vs. automation.
- **Playtesting with AI Agents:** Discuss how AI playtesting agents identify design flaws before human testing and the limitations of AI vs. human player experience.
- **Narrative AI:** Build branching dialogue using AI-generated scripts; critically reflect on authorship, narrative coherence, and player agency in AI-assisted storytelling.
- **Responsible Game Design:** Critically analyze AI-driven monetization mechanics (dynamic pricing, targeted loot boxes) and discuss ethical design obligations.

Applications:

- Video Game Design and Development Studios
- Serious Games for Education, Healthcare, and Military Training
- Gamification of Enterprise Applications and Learning Platforms
- Interactive Narrative and Transmedia Experience Design
- Board Game and Tabletop Game Design and Publishing

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

- Analyze and deconstruct core mechanics of 3 different game genres (platformer, puzzle, RPG).
- Create a paper prototype of a 2-player board game; conduct a structured playtest session with documented feedback.
- Design a complete level layout for a 2D platformer annotating challenge points, rewards, and tutorial flow.
- Use Twine to build a 3-scene branching narrative story with at least 3 distinct endings.
- Design a balanced resource economy for a strategy game; verify balance using a spreadsheet simulation.
- Create 3 distinct character concept profiles (protagonist, antagonist, supporting) with full backstory.
- Conduct a guided playtesting session for a peer's game prototype; write a structured feedback report.
- Design a progression system (XP, skill tree) for an RPG mapped to player psychology principles.
- Create a complete UI/UX wireframe for a mobile game HUD using Figma.
- Write a comprehensive 15-page Game Design Document (GDD) for an original game concept.

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the real software engineering problems.

Project 1 – Original Board Game Design

Problem: Design, prototype, and playtest a complete 2–4 player board game with unique mechanics and balance.

Concepts used: Game mechanics, Economy design, Paper prototyping, Playtesting

Industry relevance: Tabletop game publishing and educational game design.

Project 2 – Interactive Narrative Game

Problem: Branching narrative adventure game in Twine with complex choices, multiple endings, and rich world-building.

Concepts used: Narrative design, World-building, Twine, Character design

Industry relevance: Interactive fiction and visual novel studios.

Project 3 – Complete GDD for Digital Game

Problem: Professional-quality Game Design Document for an original video game concept with all systems documented.

Concepts used: GDD writing, Systems design, Level design, UI/UX design

Industry relevance: Game industry pitching, indie development, graduate portfolios.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand and apply foundational game design theories including MDA framework, flow theory, and player psychology.	L2
2	Design engaging game mechanics, balanced systems, and progression structures with documented design rationale.	L3
3	Apply level design principles to create well-paced, accessible, and discovery-driven game spaces.	L3
4	Develop compelling narrative designs and world-building documents with coherent story structures and characters.	L4
5	Create professional GDDs, conduct and analyze playtests, and iterate game designs based on evidence.	L5

Table: Mapping Levels of COs to POs / PSOs													
	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	2	2	2	2	3						2	2	3
CO2	2	3	3	3	2						2	3	2
CO3	2	3	3	3	2						2	3	2
CO4	2	3	3	3	3						2	3	3
CO5	2	3	3	3	3						3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools:

- Twine – Interactive Narrative Tool
- Figma – UI/UX Design and Prototyping
- GDevelop – No-code 2D Game Prototyping
- Miro / Mural – Collaborative Design Boards
- Notion / Google Docs – GDD Writing
- Canva / Midjourney – Concept Art Generation

TEXT BOOKS:

1. Jesse Schell, The Art of Game Design: A Book of Lenses, 3rd Ed, CRC Press, 2020.
2. Richard Rouse III, Game Design: Theory and Practice, 2nd Ed, Wordware, 2005.

3. Scott Rogers, Level Up! The Guide to Great Video Game Design, 2nd Ed, Wiley, 2014.

REFERENCE BOOKS:

1. Brenda Romero & Ian Schreiber, Challenges for Game Designers, Cengage Learning, 2009.
2. Ernest Adams, Fundamentals of Game Design, 3rd Ed, New Riders, 2014.

E-RESOURCES:

- https://gdcvault.com (GDC Vault)
- https://www.youtube.com/@extracredits (Extra Credits)
- https://gamedesignconcepts.wordpress.com
- https://learn.nvidia.com (NVIDIA DLI Free Courses)

Activity Based Learning (Suggested Activities in Class)

- Weekly game analysis assignments – deconstruct design pillars of assigned games.
- Design jams – 2-hour rapid game concept creation in teams of 3 students.
- Guest lectures from game designers at local indie studios or game companies.
- Game exhibition – students play and critique each other's Twine narrative games.

****

Quantum Algorithms

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30 hrs L + 15 hrs J
L–T–P–J	:	2–0–0–2			

Course Vision: The course aims to develop a strong foundation in quantum algorithms and quantum computation for solving complex computational problems beyond the capabilities of classical systems. It enables students to understand and implement major quantum algorithms, quantum search techniques, and error correction methods using modern quantum computing frameworks such as Qiskit. The course also prepares learners to explore advanced applications of quantum computing in optimization, cryptography, data processing, and next-generation intelligent computing systems.

Prerequisites: Python Programming, Basics of Quantum Computing

Course Learning Objectives:

This Course will enable students to:

1. Understand principles of quantum algorithms and computation.
2. Analyze major quantum algorithms like Deutsch–Jozsa, Grover's, and Shor's.
3. Explore quantum search techniques and computational complexity.
4. Study quantum noise models and error correction methods.
5. Implement quantum algorithms using Qiskit and evaluate performance.

Broad Course Description

The course *Quantum Algorithms* provides a comprehensive understanding of the principles and design of quantum algorithms that offer computational advantages over classical methods. It covers fundamental concepts such as quantum interference and parallelism, followed by key algorithms including Deutsch, Deutsch–Jozsa, Simon's, Grover's, and Shor's algorithms. The course further explores quantum search techniques, oracle models, and computational limits. It also addresses practical challenges such as quantum noise and error correction. Using Qiskit, students gain hands-on experience in implementing and analyzing quantum algorithms for real-world applications in optimization, cryptography, and data processing.

UNIT – I	08 Hours

Quantum Algorithmic Principles:

Hadamard Gates, Phase Gate, Matrix Representation of Serial and Parallel Operations, Quantum Interference, Quantum Parallelism, Function Evaluation.

(Text Book-1: Chapter 9)

UNIT – II	10 Hours

Quantum Algorithms:

Deutsch's Algorithm, Deutsch–Jozsa Algorithm, Simon's Algorithm, Grover's Search Algorithm, Shor's Factoring Algorithm

(Text Book-2: Chapter 4)

UNIT – III	09 Hours

Quantum Search Algorithms:

Quantum Search Algorithm, Oracle Model, Algorithm Procedure, Geometric Visualization, Quantum Counting, NP-Complete Problems, Unstructured Search, Optimality and Black Box Limits

(Text Book 3: Chapter 6)

UNIT – IV	09 Hours

Quantum Noise and Error Correction: Single-Qubit Errors, Kraus Operators, Depolarization Channel, Bit Flip and Phase Flip Errors, Amplitude Damping, Phase Damping, Quantum Error Correction. (Text Book1: Chapter 12)	
UNIT – V	**09 Hours**
Implementation using Qiskit: Quantum Circuit Implementation, Simulation of Algorithms, Execution on Quantum Simulators, Performance Evaluation	

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **Quantum-Enhanced Intelligent Computation:** Leverage quantum principles such as superposition, interference, and parallelism to design AI systems capable of solving complex problems more efficiently than classical methods.
- **Algorithmic Intelligence using Quantum Models:** Apply quantum algorithms (Deutsch, Grover, Shor) to enhance search, optimization, and decision-making processes in AI applications.
- **Efficient Search & Optimization:** Utilize quantum search techniques and amplitude amplification to improve performance in large-scale AI problems such as pattern recognition and database search.
- **Noise-Aware AI Systems:** Incorporate quantum noise models and error correction techniques to ensure reliability, robustness, and accuracy in quantum AI computations.
- **Simulation-Driven AI Development:** Use Qiskit-based simulators to design, test, and validate quantum algorithms, enabling reproducible and scalable AI solutions.
- **Responsible & Explainable Quantum AI:** Ensure transparency, fairness, and ethical deployment of quantum algorithms by analyzing their impact, limitations, and computational boundaries.

Suggested Tools:

- Qiskit
- PennyLane
- Google Cirq
- Google Colab

Applications:

1. Quantum search for large-scale data retrieval and recommendation systems
2. Cryptography and cybersecurity using Shor’s algorithm
3. Optimization problems in logistics, scheduling, and supply chains
4. Drug discovery and molecular simulation
5. Financial modeling and portfolio optimization
6. AI-based decision systems with quantum speedup
7. Pattern recognition and anomaly detection using quantum algorithms

Mini Project Component

Industry-Driven Projects (IBM / Google Quantum Style)

These projects simulate real-world **quantum algorithm and AI problem-solving scenarios**.

Project 1 – Quantum Search Engine

1. **Problem**: Design a search system using Grover’s algorithm
2. **Concepts used**: Oracle model, amplitude amplification
3. **Industry relevance**: Search engines, recommendation systems

Project 2 – Noise-Aware Quantum Algorithm Simulator

1. **Problem:** Analyze the effect of noise and apply error correction
2. **Concepts used**: Kraus operators, noise channels, QEC
3. **Industry relevance**: Reliable quantum computing systems

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain quantum algorithmic principles and gate operations.	L2
2	Apply quantum algorithms to solve computational problems	L3
3	Analyze quantum search algorithms and their efficiency.	L4
4	Evaluate quantum noise and error correction techniques.	L5
5	Design and implement quantum algorithms using Qiskit.	L6

Table: Mapping Levels of COs to POs / PSOs													
COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2			2						2	2	
CO2	3	3	2	2	3						2	3	2
CO3	2	3	2	3	2						2	3	2
CO4	2	2		2	2	2	2	2			2	2	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. David McMahon – *Quantum Computing Explained*, WILEY-INTERSCIENCE A John Wiley & Sons, Inc., Publication
2. Noson S. Yanofsky, Mirco A. Mannucci – *Quantum Computing for Computer Scientists*, Cambridge University Press
3. Michael A. Nielsen & Isaac L. Chuang, "Quantum Computation and Quantum Information" CAMBRIDGE UNIVERSITY PRESS Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore

REFERENCE BOOKS:

1. Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia – Programming Quantum Computers, O'Reilly Media, 2019.
2. Maria Schuld and Francesco Petruccione – Machine Learning with Quantum Computers, Springer, 2018.

E-Resources:

- https://qiskit.org
- https://pennylane.ai
- https://quantum.ibm.com
- https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Simulation of quantum gates (Hadamard, Phase) using Qiskit
Activity 2: Implementation of Deutsch–Jozsa and Grover's algorithms
Activity 3: Visualization of quantum search using geometric interpretation
Activity 4: Analysis of quantum parallelism and interference through examples

Activity 5: Simulation of quantum noise and error correction techniques

****

Business Management Using Cloud

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:			03
Hours / Week	:	03 Hours	**Total Hours**:	45Hrs L
L–T–P-J	:	3–0–0-0		

Course Vision: The course aims to develop knowledge and practical skills in cloud computing for enabling digital business transformation and modern enterprise management. It equips students with the ability to understand cloud platforms, enterprise applications, and cloud-based business solutions for improving scalability, efficiency, and innovation. The course also emphasizes cloud governance, security, cost optimization, and data-driven decision-making required to design and manage sustainable cloud-enabled business strategies in modern organizations.

Prerequisites: Basics of Computer Applications, Fundamentals of Management, Introduction to Cloud Computing (preferred)

Course Learning Objectives:

This Course will enable students to:

1. Understand cloud computing concepts and their role in business transformation.
2. Apply cloud platforms for business operations and decision-making.
3. Analyze cloud-based enterprise solutions and services.
4. Evaluate security, cost, and governance in cloud adoption.
5. Develop cloud-enabled business strategies and digital solutions

Broad Course Description

The course focuses on leveraging cloud computing technologies for modern business management and digital transformation. It introduces students to cloud service models, deployment strategies, and enterprise applications such as ERP, CRM, and analytics platforms delivered via the cloud. Students will explore how organizations use cloud platforms like Amazon Web Services, Microsoft Azure, and Google Cloud to optimize operations, enhance scalability, and drive innovation. The course also integrates concepts of cost optimization, cloud governance, data-driven decision-making, and security. Through case studies and practical exposure, students will develop skills required to manage and innovate cloud-based business systems.

UNIT – I	09 Hours
Introduction to Cloud Computing in Business: Cloud computing fundamentals, Service models: IaaS, PaaS, SaaS, Deployment models: Public, Private, Hybrid, Role of cloud in digital transformation, Business benefits and challenges *(Text Book 1: Chapters 1,2 | Ref 2)*	
UNIT – II	**09 Hours**
Cloud Platforms & Business Applications: Overview of AWS, Azure, Google Cloud, Cloud-based ERP, CRM, HRM systems, SaaS tools for business operations, Cloud-based collaboration tools Case studies of cloud adoption *(Text Book 2: Chapters 3,4 | Ref 1,3)*	
UNIT – III	**09 Hours**

Cloud Data Management & Analytics: Data storage and databases in cloud, Big data and business analytics, Cloud-based BI tools, Data-driven decision-making, Introduction to AI in cloud business *(Text Book 3: Chapters 5,6 | Ref 2,4)*	
UNIT – IV	**09 Hours**

Cloud Security, Governance & Costing: Cloud security fundamentals, Identity and access management (IAM) , Data privacy and compliance, Cloud governance frameworks, Cost models and optimisation strategies *(Text Book 1: Chapter 7 | Ref 3,4)*	
UNIT – V	**09 Hours**
Cloud Strategy & Digital Business Transformation: Cloud migration strategies, Business continuity and disaster recovery, DevOps and cloud integration , Digital transformation frameworks, Future trends: Serverless, Edge, AI-driven cloud *(Text Book 2,3 | Ref 1,4)*	

AI Integration & Digital Intelligence

- Business analytics using Google Analytics
- Cloud dashboards using Power BI
- AI services via Amazon Web Services AI tools
- Automation using Zapier
- Collaboration using Microsoft Teams

Applications

1. Cloud-based enterprise resource planning
2. Digital business transformation
3. E-commerce platforms
4. Financial analytics systems
5. Remote business operations

Project Component

Project 1 — Cloud-Based Business Solution

- Develop a cloud-enabled business model (e.g., CRM system)
- Tools: AWS / Azure

Project 2 — Cloud Cost Optimisation Analysis

- Analyse cost models and optimise cloud usage. Deliverables: Report + dashboard

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain cloud computing concepts and business applications	L2
CO2	Apply cloud platforms for business operations	L3
CO3	Analyze cloud-based enterprise solutions	L4
CO4	Evaluate security, governance, and cost factors	L5
CO5	Design cloud-enabled business strategies	L6

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2								2		3	2
CO2	2	3	3		2					3		2	3
CO3	2	2	3	2	3					2		2	2
CO4	2	3	2	3	2	2		2		2		2	3
CO5	3	3	3	2	3	2		2	3	3	2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Cloud Computing: Concepts, Technology & Architecture, Thomas Erl, Prentice Hall, 1st Edition, 2013.
2. Cloud Strategy, Gregor Hohpe, O'Reilly Media, 1st Edition, 2021.
3. Architecting the Cloud, Michael J Kavis, Wiley, 1st Edition, 2014

REFERENCE BOOKS:

1. The Lean Startup, Eric Ries, Crown Business, 2011.
2. CompTIA Cloud+ Study Guide, Todd Montgomery, Stephen Olson, Sybex, 2nd Edition, 2018.
3. AWS Certified Solutions Architect Official Study Guide, Ben Piper, David Clinton, Wiley, 2nd Edition, 2019.

E-RESOURCES:

1. NPTEL – Cloud Computing Courses
2. Coursera – Cloud & Business Analytics
3. edX – Digital Transformation
4. Amazon Web Services Training Portal
5. Microsoft Azure Learning

Cloud Industry Mapping (Alignment)

Platform / Certification	Mapped Units	Relevance
AWS Cloud Practitioner	Unit I, II	Cloud fundamentals and services
Azure Fundamentals	Unit I, IV	Governance, security, compliance
Google Cloud Digital Leader	Unit III, V	Analytics and business transformation
AWS Solutions Architect	Unit II, IV	Architecture and cost optimisation

Activity-Based Learning

1. Cloud service comparison (AWS vs Azure vs GCP)
2. Hands-on SaaS tool demonstrations

3. Case study analysis (Netflix, Amazon cloud adoption)
4. Cost estimation exercises
5. Group project on digital transformation

****

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Modern Computer Vision, IIT Madras [NPTEL] 12 WEEKS
2	*CID	Cryptography and Network Security, IIT Kharagpur [NPTEL] 12 WEEKS
3	*CID	Quantum Computing: Algorithms and Limitations Through the Query Model [NPTEL] 12 WEEKS

Note: CID = NPTEL Course ID

Minor Project

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	02
Hours / Week	:	04 Hours	**Total Hours**	:	30 Hours J
L–T–P-J	:	0-0-0-4			

Course Vision:

Course Learning Objectives:

This Course will enable students to:

1. To apply theoretical knowledge to solve practical problems.
2. Develop professional project management and communication skills.
3. Able to work in teams and present the project work.

Guidelines for Minor Project

1. Each B.Tech Project must be carried out by a group of students at the Institute. To ensure uniform participation of each student, the group size should be preferably at least 3 but not more than 4 students.
2. Each project activity must be supervised by the faculty members of the Institute. These faculty members are termed Project Guides.
3. In case the project is of multi-disciplinary nature, the Project group can be formed consisting of students from other Departments. But there must be at least one student and a project Guide from the Department who is offering the Project.
4. The topic proposed by both the guide and the student team should be approved by the department chairman and the department project coordinator to proceed further. A degree of industrial input and involvement will be encouraged and can be facilitated through existing academic-industrial collaborations or by addressing specific topics that are of interest to industrial partners.
5. All projects will be closely supervised by the Project Guide with ongoing feedback and guidance at all stages of the project from conception to completion.
6. The following criteria will be checked by the department chairman to approve for the project proposal:
   a. Department staff as Project guide
      i. Ability to provide direction to the student in the chosen field of interest to formulate a suitable title of the project.
      ii. Ability to design an appropriate strategy and methodology to carry out the Project by the team.
      iii. Ability to provide and evaluate the strong literature review document for the chosen topic.
      iv. Ability to train students on paper / technical writing skills
   b. Student Team
      i. To be dedicated and committed to work on the project by sharpening the existing and learning new technical skills.
      ii. To be committed to completing the project and participate in hackathons and project exhibitions.
7. Phase-1 comprises of Literature Survey, Problem identification, Objectives and Methodology.
8. There will be CIA evaluation (Project reviews) done by a committee of senior faculty of the Department based on the rubrics

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Augmented Problem-Solving:** Use AI tools for requirement analysis, design suggestions, coding assistance, and automated testing.
- **Evidence-Based Decision-Making with AI Support:**

Apply AI dashboards, simulations, and optimization engines to assess feasibility, effectiveness, and efficiency.

- **Critical Evaluation of AI Bias & Limitations**: Encourage questioning of AI outputs, identification of biases, and validation against authentic evidence.
- **Ethical & Responsible AI Use**: Integrate Responsible AI principles (fairness, accountability, transparency) into problem-solving and evaluation tasks.
- **Human–AI Collaboration for Creativity & Innovation**: Position AI as a co-creator in solution design and evaluation, while human intelligence ensures originality.
- **Sustainability & Societal Impact Awareness**: Link AI-driven solutions to Sustainable Development Goals (SDGs) for real-world relevance.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Demonstrate practical problem-solving skills by analyzing requirements, designing solutions, implementing algorithms, and testing software systems in real-world scenarios.	L4
CO2	Develop critical thinking skills, enabling them to evaluate the feasibility, effectiveness, and efficiency of proposed solutions and make informed decisions based on evidence and reasoning.	L5

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	3	3	3	3	3	3	3	3	3	3	3	3
CO2	3	3	3	3	3	3	3	3	3	3	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

****

Small E- Business Launch

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	02 Hours	**Total Hours**	:	30Hrs L+15 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop entrepreneurial and digital business skills required to identify market opportunities, validate innovative ideas, and launch sustainable e-business ventures. It enables students to understand customer needs, business models, feasibility analysis, and digital trade practices for building successful online businesses. The course also equips learners with practical knowledge in lean start-up methodologies, digital platforms, business planning, and legal aspects necessary for managing modern technology-driven entrepreneurial ventures.

Course Learning Objectives:

This Course will enable students to:

1. To introduce students to the fundamentals of entrepreneurship and digital business transformation.
2. To develop the ability to identify customer needs and market opportunities.
3. To equip learners with skills for idea generation, validation, and feasibility analysis.
4. To understand business models, strategies, and planning techniques.
5. To familiarize students with legal, operational, and digital trade aspects of entrepreneurship.

Broad Course Description

The course Small E-Business Launch provides a comprehensive understanding of the entrepreneurial process in the digital era, focusing on identifying customer needs, generating and validating business ideas, and developing viable digital business opportunities. It covers essential aspects such as business models, feasibility analysis, and lean start-up methodologies, along with practical skills in wireframing, service-market fit, and logistics. The course also emphasizes legal considerations, digital platforms, and emerging practices like virtual trade shows, enabling students to design, launch, and manage sustainable e-business ventures using modern tools and strategies.

UNIT – I	**06 Hours**
INTRODUCTION: Introduction, Identifying Customer Needs, Opportunity Identification & Planning, Idea Generation and Validation, Entrepreneurial Culture, Digital Business Opportunity.	
UNIT – II	**07 Hours**
Business Plan & its Components, Business Model and Strategies, Selection of Appropriate Business Model, Business Model Canvas, Feasibility.	
UNIT – III	**06 Hours**
Lean thinking, Specifications, Wireframe Generation and Validation.	
UNIT – IV	**07 Hours**
Service-Market Fit, Logistics of Goods Delivery, Virtual Trade Shows	

UNIT – IV	**04 Hours**
Legal Aspects & Entrepreneurial Law	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-driven Opportunity Identification:** Use AI tools for customer need analysis and idea validation, while ensuring ethical data usage and unbiased decision-making.
- **Intelligent Business Modelling:** Apply AI-based analytics for Business Model Canvas and feasibility studies with transparent and explainable assumptions.
- **Lean AI Prototyping:** Integrate AI tools for wireframing, rapid prototyping, and validation, ensuring sustainable and resource-efficient development.
- **AI-enabled Market Fit Analysis:** Use AI for predicting service-market fit and customer behaviour, with a focus on privacy-aware and responsible data handling.
- **Smart Logistics & Digital Platforms:** Implement AI in logistics optimization and virtual trade systems while promoting fairness, security, and sustainability in digital operations.

Suggested Tools:

- Perplexity AI
- Shopify
- Canva Magic Design
- Mailchimp AI
- Google Analytics

Applications:

1. Analyze customer needs and behaviour using AI while ensuring ethical and privacy-aware data usage.
2. AI-powered Business Model Optimization.
3. AI-driven Prototyping & Wireframing Tools.
4. AI-enabled Market Prediction Systems.
5. AI in Smart Logistics & Supply Chain.

Project Component

- **Healthcare & Wellness:** Online consultation, fitness platforms, mental health support, pharmacy delivery.
- **Education & EdTech:** E-learning platforms, skill development portals, tutoring services.
- **Food & Hospitality:** Online food delivery, cloud kitchens, restaurant booking systems.
- **Logistics & Supply Chain:** Delivery services, inventory management, smart logistics solutions.
- **FinTech & Digital Payments:** Budgeting apps, digital wallets, financial advisory platforms.
- **Agriculture & Rural Tech:** Farm-to-market platforms, agri-product sales, farmer support systems.

- **Sustainable & Green Business:** Eco-friendly products, recycling platforms, sustainable services.
- **Service-Based Platforms:** Home services (cleaning, repair), freelancing, booking systems.
- **Social & Community Platforms:** Local business promotion, event management, community marketplaces.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Identify customer needs, analyze opportunities, and generate validated ideas for digital business applications.	L4
CO2	Create a business plan and design appropriate business models using tools like Business Model Canvas and feasibility analysis.	L6
CO3	Apply lean thinking principles to design, prototype, and validate solutions through specifications and wireframes.	L3
CO4	Analyze logistics and implement and present a sustainable e-business solution	L4
CO5	Evaluate service–market fit and legal aspects required for developing a sustainable e-business solution.	L5

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	3	2			2	2	2	2	3
CO2	3	3	3	2	3	2			2	3	3	2	3
CO3	2	3	3	2	3	2			2	2	2	2	3
CO4	2	3	3	3	3	3	2	2	2	3	3	2	3
CO5	2	2	2	2	2	2	2	2	2	2	2		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Fisher, Steven, Duane, Ja-Nae (2016). Start-up Equation A Visual Guidebook for Building your Start- up, New Delhi: Tata McGraw-Hill.
2. Ulrich, K. T. (2019). Product design and development. Tata McGraw-Hill Education, 7th edition.

REFERENCE BOOKS:

1. Holt H., David (2017). Entrepreneurship: New Venture Creation, New Delhi: Pearson Education.

2. Zimmerer, W. Thomas and Scarborough, M. Norman and Doug Wilson (2009). Essentials of Entrepreneurship and Small Business Management, 5/e; New Delhi: Prentice Hall India.

E-RESOURCES:

1. https://onlinecourses.nptel.ac.in/noc24_mg92/preview
2. https://alison.com/tag/business-management
3. https://www.coursera.org/specializations/start-your-own-business
4. https://www.coursera.org/projects/facebook-small-business-marketing
5. https://www.udemy.com/course/launching-your-small-business-in-india/

Activity Based Learning (Suggested Activities in Class)

1. Proposal Handout & Presentation

a. The Problem
b. Need
c. Existing Alternatives
d. Potential Market

2. Wireframe Sketches, Target Specifications

a. List Critical Customer Needs
b. For a., Prepare a List of Target Specifications
c. Prepare Wireframe

****

Lean Startup Methodology

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:		**Credits**	:	03
Hours / Week	:	02 Hours	**Total Hours**	:	30Hrs L+15 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop entrepreneurial thinking and innovation skills required to transform ideas into sustainable and scalable start-up ventures using Lean Start-up methodologies. It enables students to identify market opportunities, validate solutions through experimentation and customer feedback, and design effective business models using practical frameworks and strategic tools. The course also equips learners with problem-solving, decision-making, customer acquisition, and business growth skills necessary for building and managing innovative start-ups in dynamic market environments.

Course Learning Objectives:

This Course will enable students to:

1. Analyze the opportunities and identify the potential problem to build a solution
2. Devise the solution to solve the given problem and build core PoC and business model.
3. Explain the business model iterations based on the Field experience with customers.
4. Get the idea of acquiring / retaining customers as well basic finance aspects.
5. Describe and illustrate the current and future path forward of your (potential) start-up.

Broad Course Description

This course introduces students to the Lean Start up Methodology, helping them understand how to turn ideas into successful businesses through continuous learning and experimentation. It follows three stages: Vision (start, define, learn, and experiment)**,** where students identify problems and explore ideas; Steer (leap, test, measure, and pivot)**,** where they validate ideas using MVPs and feedback; and Accelerate (batch, grow, adapt, and innovate)**,** where they scale and improve their business models. The course also covers Business Model development, including the 9 building blocks and Business Model Canvas, along with design aspects like customer insights, ideation, prototyping, and storytelling, as well as strategy and process for evaluating and managing business models. Overall, the course helps students develop the skills to build, test, and grow innovative start up ideas in a simple and practical way.

UNIT – I	06 Hours
VISION: Start, Define, Learn & Experiment.	
UNIT – II	**06 Hours**
STEER: Leap, Test, Measure & Pivot	
UNIT – III	**06 Hours**
ACCELERATE: Batch, Grow, Adapt, Innovate	

UNIT – IV	06 Hours
BUSINESS MODEL PART I Business Model Definition, 9 Building Blocks, Business Model Canvas, Patterns	

UNIT – V	06 Hours
BUSINESS MODEL PART II Design – Customer insights, Ideation, Prototyping, Story Telling Strategy – BM environment, Evaluating BM's, Manage Process – Business Model Design Process	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Use AI tools to analyze customer needs, market gaps, and trends while ensuring ethical data usage, avoiding bias, and maintaining transparency in idea validation.
- Apply AI-driven insights to design Business Model Canvas and evaluate feasibility with explainable and data-backed assumptions.
- Use AI tools for rapid prototyping and MVP development while ensuring efficient use of resources and sustainable design practices.

Suggested Tools:

- Google Trends
- Crunch base
- Uizard

Applications:

- Wireframing and UI design
- MVP creation
- Rapid testing and iteration
- Cohort and retention analysis

Project Component

- Develop a system that uses AI to analyze customer reviews, social media data, and trends to validate start up ideas and suggest improvements.
- Build a platform that predicts product success by analyzing user behaviour, engagement metrics, and feedback using AI techniques.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain the principles of the Lean Startup methodology and its application in early-stage business development.	L2
CO2	Develop a Minimum Viable Product (MVP) using validated learning and measurement techniques.	L3
CO3	Analyze growth engines and strategies for scaling startups sustainably.	L4
CO4	Construct a Business Model Canvas incorporating all nine building blocks.	L3
CO5	Evaluate business models using design thinking, prototyping, and strategic assessment.	L5

Table: Mapping Levels of COs to POs / PSOs

CO S	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO1 1	PSO 1	PSO 2
CO1	3	3	2	2	3	2		2	2	2	2		3
CO2	3	3	3	2	3	2		2	2	3	3		3
CO3	2	3	3	2	3	2		2	2	2	2		3
CO4	2	3	3	3	3	3		2	2	3	3		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Eric Ries "The Lean Startup", ISBN–978-0670921607 https://ia801206.us.archive.org/31/items/TheLeanStartupErickRies/The%20Lean%20 Startup%20-%20Erick%20Ries.pdf
2. Alex Osterwalder, Yves Pigneur "Business Model Generation Handbook" ISBN-978-0-470-87641-1 http://alvarestech.com/temp/PDP2011/pdf/Business%20Model%20Generation%20(1). pdf

REFERENCE BOOKS:

1. Steve Blank "The Four Steps to the Epiphany: Successful Strategies for Products That Win", ISBN-13 : 978-0989200509

2. The Entrepreneur's Guide to Customer Development: A "Cheat Sheet" to the Four Steps to the Epiphany Paperback – Import, 6 February 2012, **ISBN-13 :** 978-0982743607

E-RESOURCES:

1. https://ia801206.us.archive.org/31/items/TheLeanStartupErickRies/The%20Lean%20 Startup%20-%20Erick%20Ries.pdf
2. http://alvarestech.com/temp/PDP2011/pdf/Business%20Model%20Generation%20(1).pdf
3. https://www.getstoryshots.com/books/the-lean-startup-summary/
4. https://www.strategyzer.com/canvas

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving with on field exposure
2. Build Start-up and test the solution with real customers.

****

Entrepreneurial Finance & Deep-Tech Start-ups

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**:	02
Hours / Week	:	02 Hours	**Total Hours**:	30 Hrs L
L–T–P-J	:	2–0–0-0		

Course Vision:

The course aims to develop entrepreneurial, financial, and technological skills required to build and scale AI-driven deep-tech start-ups in emerging innovation ecosystems. It enables students to understand start-up finance, valuation methods, funding strategies, and AI-powered business decision-making for technology ventures. The course also equips learners with practical knowledge in financial modelling, market analysis, venture pitching, and ethical innovation to create sustainable and scalable solutions in domains such as AI, robotics, IoT, and advanced digital technologies.

Prerequisites: - Basics of Economics, AI/ML Fundamentals, Data Science

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamentals of entrepreneurial finance and start-up ecosystems.
2. Analyse valuation methods and funding strategies for AI-driven start-ups.
3. Apply AI tools for financial modelling, forecasting, and market analysis.
4. Develop skills to design, pitch, and scale deep-tech ventures.
5. Evaluate ethical, regulatory, and societal implications of AI entrepreneurship.

Broad Course Description

Entrepreneurial Finance & AI-Driven Deep-Tech Start-ups is an interdisciplinary course that integrates financial literacy, entrepreneurship, and advanced technology domains such as Artificial Intelligence and deep-tech innovation. The course equips students with the knowledge and skills required to build, evaluate, and scale start-ups in emerging technology sectors. Students will explore start-up ecosystems, financial planning, valuation techniques, and funding mechanisms including venture capital, angel investment, and government grants. The course emphasizes AI-driven decision-making for market analysis, forecasting, and business strategy development. It also introduces business models tailored for AI and deep-tech domains such as robotics, IoT, and quantum technologies. Learners will engage in hands-on activities including financial modelling, pitch deck creation, and start-up simulations using modern tools like Excel, Power BI, and AI platforms. Ethical considerations, legal frameworks, intellectual property rights, and sustainability in AI-driven entrepreneurship are integral components of the course. By the end of the course, students will be capable of conceptualizing, validating, and presenting scalable and responsible deep-tech start-up ideas.

UNIT – I	**05 Hours**

Foundations of Entrepreneurial Finance

Start-up ecosystem overview, types of start-ups (SaaS, AI, Deep-Tech), financial statements (Profit & Loss, Balance Sheet, Cash Flow), unit economics (Customer Acquisition Cost, Lifetime Value), break-even analysis.

UNIT – II	**05 Hours**

Start-up Valuation & Funding Strategies

Start-up valuation methods (DCF, Comparable, Scorecard), capitalization tables, equity vs debt financing, funding stages (Seed, Series A/B/C), venture capital, angel investors, government funding schemes.

UNIT – III	**05 Hours**

AI-Driven Business Models & Deep-Tech Ventures AI business models (platform, API-based, subscription), deep-tech domains (AI, robotics, IoT, quantum computing), product-market fit, competitive analysis, monetization strategies	
UNIT – IV	**05 Hours**
Financial Planning, Forecasting & Scaling Financial projections, burn rate, runway, scenario planning, scaling strategies, growth metrics, exit strategies (IPO, M&A).	
UNIT – V	**05 Hours**
Legal, Ethical & Future of AI Entrepreneurship Start-up legal frameworks, intellectual property (patents, copyrights), AI regulations, data privacy laws, ESG principles, future trends in AI entrepreneurship	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Augmented Financial Analysis:** Use AI tools for financial dashboards, forecasting, and automated analysis.
- **AI-Assisted Start-up Ideation:** Leverage generative AI for idea validation, market research, and MVP design.
- **Human vs AI Decision Making:** Compare human intuition vs AI-driven insights in financial and strategic decisions.
- **Data-Driven Entrepreneurship:** Use AI tools for market intelligence and predictive analytics.
- **Ethical AI Entrepreneurship:** Address bias, fairness, and responsible innovation in start-up ecosystems.
- **Experiential Learning:** Build real-world AI start-up projects with AI-assisted tools.

Applications:

1. AI-based start-up financial planning
2. Venture capital investment analysis
3. Deep-tech innovation ecosystems
4. FinTech and AI-driven business models
5. Start-up valuation and fundraising
6. Smart city and healthcare start-up solutions

Mini Project Component

Industry-Driven Projects

Project 1 – AI Startup Financial Model

- Problem: Build financial projections for an AI-based startup.
- Concepts: Unit economics, forecasting, valuation
- Industry relevance: Startup financial planning

Project 2 – Deep-Tech Startup Pitch

- Problem: Develop and pitch a scalable AI startup idea
- Concepts: Business model, funding strategy, AI integration
- Industry relevance: Venture capital ecosystem

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Apply financial concepts such as cash flow and valuation	L3
CO2	Design AI-driven business models for deep-tech ventures	L3
CO3	Analyze startup funding strategies	L4
CO4	Develop financial projections and pitch decks	L6
CO5	Build scalable and ethical AI startups	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	2	2				3	3	3	3	2
CO2	3	3	2	2	3	2			3	3	3	3	2
CO3	3	3	2	3	3	2			3	3	3	3	2
CO4	2	3	2	3	3	2			3	3	3	3	2
CO5	2	3	2	3	3	2			3	3	3	3	2

3: Substantial (High)   2: Moderate (Medium)   1: Poor (Low)

Suggested Tools

- Excel, Power BI, Tableau
- Python, SQL
- AI Tools (ChatGPT, AutoML, forecasting tools)
- Startup tools (Pitch deck, Cap table tools)

TEXT BOOKS:

1. Brad Feld & Jason Mendelson – *Venture Deals*
2. Eric Ries – *The Lean Start-up*

REFERENCE BOOKS:

1. Ash Maurya – *Running Lean*
2. Scott Kupor – *Secrets of Sand Hill Road*
3. Harvard Business Review – Entrepreneurship Articles
4. NASSCOM & Start-up India Reports

E-RESOURCES:

1. Coursera – Entrepreneurship & Finance
2. edX – Start-up & Innovation Courses
3. NPTEL – Entrepreneurship

Activity Based Learning (Suggested Activities in Class)

1. Start-up Pitch Simulation

2. Financial Modelling Workshop
3. Case Study Discussions (AI Start-ups)
4. Ethical AI Entrepreneurship Debate

****

AI in Healthcare

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	04
Hours / Week	:	04 Hours	**Total Hours**	:	45Hrs L +15Hrs J
L–T–P-J	:	3–0–0-2			

Course Vision:

The course aims to develop strong foundations in Artificial Intelligence for Healthcare by integrating machine learning, healthcare analytics, and intelligent decision-support systems for improving patient care and clinical outcomes. It enables students to understand healthcare data systems, medical AI applications, predictive modelling, and advanced techniques such as deep learning and explainable AI for real-world healthcare solutions. The course also promotes ethical, secure, and responsible AI practices required for developing reliable and human-centered healthcare technologies in modern medical environments.

Prerequisites: Machine Learning, Deep Learning

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamental concepts of AI in healthcare, healthcare data systems, and digital health technologies.
2. To analyze healthcare data sources, data collection methods, and preprocessing techniques for clinical datasets.
3. To design and apply machine learning models for healthcare applications such as diagnosis, prediction, and decision support systems.
4. To evaluate AI techniques in medical imaging, natural language processing, and bioinformatics application.
5. To apply ethical, explainable, secure, and regulatory principles in the development of AI-based
6. healthcare solutions.

Broad Course Description

This course provides a comprehensive foundation in Artificial Intelligence for Healthcare, covering the complete pipeline from healthcare data acquisition and preprocessing to machine learning models and clinical decision support systems. Students will learn the principles of healthcare data analytics, AI techniques, and their applications in diagnostics, prediction, and patient care, along with modern advancements such as deep learning, explainable AI, and privacy-preserving methods. The course emphasizes real-world healthcare applications including disease prediction, medical imaging, personalized medicine, and hospital management systems, while fostering critical thinking through AI-assisted learning, ethical evaluation, and research-integrated pedagogy.

UNIT – I	06 Hours

Introduction to AI in Healthcare Systems

Evolution of Artificial Intelligence in medicine, AI paradigms in healthcare, Overview of healthcare systems and digital transformation, Healthcare data ecosystem (EHR, EMR, clinical records), Types of healthcare data (structured, unstructured, imaging, genomic), Challenges in healthcare data (heterogeneity, scalability, uncertainty), Introduction to healthcare data analytics, Role of AI in modern healthcare delivery.

(Text Book–1: Chapter 1: 1.1, 1.2, 1.3; Chapter 2: 2.1, 2.2)
(Text Book–2: Chapter 1: 1.1, 1.2; Chapter 2: 2.1)
(Text Book–3: Chapter 1; Chapter 2)
(Text Book–4: Chapter 1: 1.1, 1.2; Chapter 2: 2.1)
(Text Book–5: Relevant sections on sensing and data acquisition)

UNIT – II	06 Hours
Healthcare Data Processing and Analytics Data collection methods in healthcare, Literature search techniques (MeSH terms, PubMed, medical repositories), Healthcare databases and datasets, Data preprocessing (cleaning, normalization, missing data handling), Feature extraction and selection, Qualitative and quantitative data synthesis, Statistical analysis basics, Introduction to machine learning workflow in healthcare. (Text Book–1: Chapter 2: 2.3, 2.4; Chapter 3: 3.1, 3.2, 3.3) (Text Book–2: Chapter 3: 3.1, 3.2; Chapter 4: 4.1) (Text Book–3: Chapter 3; Chapter 4)(Text Book–4: Chapter 3: 3.1, 3.2; Chapter 4: 4.1) (Text Book–5: Relevant sections on data acquisition and processing systems)	
UNIT – III	**06 Hours**
AI Techniques and Clinical Applications Machine learning models for healthcare (classification, regression, clustering), Clinical Decision Support Systems (CDSS): architecture and types, Predictive analytics in patient care, AI in medical diagnostics, Medical imaging analysis (basic concepts), Natural Language Processing (NLP) for clinical text, AI in bioinformatics and genomics, Introduction to deep learning in healthcare. (Text Book–1: Chapter 4: 4.1, 4.2, 4.3; Chapter 5: 5.1) (Text Book–2: Chapter 5: 5.1, 5.2; Chapter 6: 6.1) (Text Book–3: Chapter 5; Chapter 6) (Text Book–4: Chapter 5: 5.1, 5.2; Chapter 6: 6.1)	
UNIT – IV	**06 Hours**
Explainable, Ethical and Secure AI in Healthcare Explainable AI (XAI) techniques and importance in clinical settings, Bias and fairness in AI models, Ethical issues in AI-driven healthcare, Data privacy and security challenges, Regulatory frameworks (HIPAA, GDPR overview), Interoperability standards in healthcare systems, Trustworthy AI and accountability, Emerging trends and future directions in AI healthcare systems. (Text Book–1: Chapter 6: 6.1, 6.2; Chapter 7: 7.1) (Text Book–2: Chapter 7: 7.1, 7.2; Chapter 8: 8.1) (Text Book–3: Chapter 7; Chapter 8) (Text Book–4: Chapter 7: 7.1, 7.2; Chapter 8: 8.1)	
UNIT – V	**06 Hours**
Case Studies and Hands-on AI in Healthcare Overview of healthcare data analytics lifecycle (based on Reddy & Aggarwal), Data understanding and exploration using real-world healthcare datasets, Data preprocessing and feature engineering in clinical data, Application of machine learning models for healthcare analytics (classification, prediction, risk scoring), Case studies on AI-driven healthcare systems (based on AI-First Healthcare concepts), AI in clinical workflow optimization, AI in hospital operations and resource management, Evaluation metrics for healthcare models (accuracy, precision, recall, ROC), Introduction to deployment considerations in healthcare AI systems, Hands-on experiments using healthcare datasets (EHR, disease prediction datasets).	
(Text Book–1: Chapter 8: 8.1, 8.2; Chapter 9: 9.1) (Text Book–2: Chapter 9: 9.1, 9.2; Chapter 10: 10.1) (Text Book–3: Chapter 9; Chapter 10) (Text Book–4: Chapter 9: 9.1, 9.2; Chapter 10: 10.1)	
• Medical AI Development with MONAI: Interactive Annotation Using NVIDIA NIM Microservices • Getting Started with NVIDIA Tools for Generative AI in Digital Health • Accelerate Data Science and Leverage Foundation Models in Digital Biology • Streamlining Drug Discovery with NVIDIA BioNeMo NIM Microservices and Blueprints • Training Healthcare Robots from Scratch With Isaac for Healthcare	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

AI-assisted analysis of healthcare systems, data ecosystems, and identification of AI opportunities in clinical workflows.

AI-based healthcare data preprocessing, feature extraction, and intelligent data synthesis for clinical datasets.

AI-driven development of clinical decision support systems, predictive models, and diagnostic tools using machine learning techniques.

AI-enabled explainability, fairness analysis, and implementation of privacy-preserving and secure healthcare AI systems.

AI-based healthcare analytics, real-world case study **modeling**, and predictive healthcare system development using datasets.

Students evaluate the limitations, risks, and ethical concerns of applying AI in healthcare systems and clinical environments.

Students analyze bias, uncertainty, and reliability issues in healthcare data and AI-driven data processing methods.

Students assess the effectiveness, interpretability, and clinical reliability of AI-based diagnostic and decision support systems.

Students examine ethical dilemmas, data privacy concerns, and regulatory challenges in deploying AI in healthcare.

Students evaluate trade-offs between model accuracy, interpretability, patient safety, and data privacy in real-world healthcare AI solutions.

Applications:

1. DiseasePrediction
2. MedicalImaging
3. ClinicalDecisionSupport
4. DrugDiscovery
5. PatientMonitoring

Mini Project Component

Some sample projects described below simulate real-world AI and healthcare system challenges.

Project 1 – AI-Based Disease Prediction System.

- Problem: Design an AI system to predict diseases (e.g., diabetes, heart disease) using patient health data.
- Concepts used: Data preprocessing, feature selection, machine learning models (classification), model evaluation metrics.
- Industry relevance: Clinical decision support systems, digital health platforms, healthcare startups, hospitals.

Project 2 – Medical Imaging Analysis System

- Problem: Develop an AI model to detect abnormalities in medical images (e.g., X-rays, MRI scans).

- Concepts used: Deep learning (CNNs), image preprocessing, pattern recognition, model training and validation.
- Industry relevance: Radiology AI systems, diagnostic centers, healthcare AI companies (e.g., imaging analytics).

Project 3 – Smart Healthcare Monitoring System

- Problem: Build an AI-enabled system to monitor patient vitals and predict health risks in real time.
- Concepts used: Time-series data analysis, anomaly detection, predictive analytics, integration with IoT/wearable data.
- Industry relevance: Remote patient monitoring, wearable healthcare devices, telemedicine platforms.

Project 4 – AI-Based Clinical Text Analysis System

- Problem: Develop a system to extract meaningful insights from clinical notes or electronic health records.
- Concepts used: Natural Language Processing (NLP), text classification, entity recognition, sentiment analysis.
- Industry relevance: Hospital information systems, EHR analytics, healthcare documentation automation.

Project 5 – Hospital Resource Optimization System

- Problem: Design an AI system to optimize hospital resources such as bed allocation, staff scheduling, and patient flow.
- Concepts used: Predictive modeling, optimization techniques, data analytics, simulation models.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Explain the fundamentals of AI in healthcare, including healthcare data systems and digital health concepts.	L2
CO2	Analyze and preprocess healthcare data using appropriate data collection, cleaning, and synthesis techniques.	L3
CO3	Design and apply machine learning models for healthcare applications such as diagnosis, prediction, and clinical decision support.	L3
CO4	Evaluate AI techniques used in medical imaging, natural language processing, and bioinformatics applications.	L3
CO5	Develop AI-based healthcare solutions considering ethical, explainable, secure, and regulatory requirements.	L3

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	2				2		2	2	2
CO2	3	2	2	2	2				2		2	2	2
CO3	3	2	2	2	2				2		2	2	2
CO4	3	2	2	2	2				2		2	2	2
CO5	3	2	2	3	2				2		2	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Textbooks:

1. C. K. Reddy and C. C. Aggarwal (Eds.), (2015), CRC Press, ISBN: 978-0367575687 Comprehensive resource covering healthcare data mining, analytics techniques, and real-world applications.
2. Kerrie L. Holley and Siupo Becker (2021), 1st Edition, O'Reilly Media, ISBN: 978-1492063155
   Focuses on AI-driven transformation in clinical workflows, hospital management, and healthcare business processes.
3. Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again Eric Topol (2019). Explores the impact of AI on healthcare delivery, doctor-patient relationships, and future medical practices.
4. Artificial Intelligence in Healthcare:Adam Bohr and Kaveh Memarzadeh (2020) Covers AI technlogies, applications, and challenges in modern healthcare systems.
5. Sensor Solutions: Driving Innovation in Industrial Automation and IoT Focus on industrial IoT, predictive maintenance, and modern sensor-driven systems.

REFERENCE BOOKS:

1. Machine Learning for Healthcare, Kevin B. Johnson, Fred S. Trotter, Nikhil G. Shah, MIT Press, 2020.
2. Artificial Intelligence for Healthcare Applications and Management, Nilanjan Dey, Amira S. Ashour, Springer, 2020.
3. Deep Learning for Healthcare,Kevin Zhou, Hayit Greenspan, Dinggang Shen, Academic Press, 2017.
4. Health Informatics: Practical Guide for Healthcare and Information Technology Professionals Robert E. Hoyt and Ann Yoshihashi, Lulu Press, 2018.
5. Biomedical Informatics: Computer Applications in Health Care and Biomedicine Edward H. Shortliffe and James J. Cimino, Springer, 2014.
6. Artificial Intelligence in Medicine A. G. Cohn, Springer (Lecture Notes / Advanced Topics).

E-RESOURCES:

1. NPTEL Courses (AI in Healthcare, Machine Learning, Data Analytics by IITs)
2. MIT OpenCourseWare (Artificial Intelligence, Machine Learning, Health Informatics)
3. Research Articles from IEEE, Elsevier, Springer on AI in healthcare, medical imaging, and health data analytics
4. Online platforms such as Coursera / edX (AI for Medicine, Healthcare Analytics courses)
5. IEEE research articles on IoT, AIoT, and sensor networks

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.E.g., Predicting disease risk using patient data, AI in early diagnosis of cancer.
2. Role play.E.g., Doctor–AI system interaction for clinical decision support, patient

data analysis workflow.

3. Demonstration of solution to a problem through programming.E.g., Implementing a simple disease prediction model using Python (classification model).
4. Flip class activity.E.g., Machine learning algorithms, data preprocessing techniques, model evaluation metrics.

****

Building LLM with Prompt engineering

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P-J	:	3–0–0-0			

Course Vision: The course aims to develop strong foundations in sensors, embedded systems, and Internet of Things (IoT) technologies for designing intelligent, connected, and secure smart systems. It enables students to understand IoT architectures, communication protocols, cloud integration, and AI-driven analytics for real-world applications such as smart cities, healthcare, and industrial automation. The course also equips learners with practical and analytical skills required to build sustainable and scalable IoT solutions using modern hardware, networking, and data-driven technologies**.**

Prerequisites: NLP with GENAI

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamental concepts of IoT architecture, M2M communication, and Industry 4.0 systems.
2. To analyze sensor and actuator technologies along with measurement and interfacing techniques.
3. To design and implement embedded systems for IoT applications using modern hardware platforms.
4. To evaluate IoT communication protocols and networking technologies for efficient data transmission.
5. To apply cloud computing, data analytics, and security principles in IoT-based solutions.

Broad Course Description

This course provides a comprehensive foundation in Sensors, Systems, and Internet of Things (IoT), covering the complete IoT stack from sensing and embedded systems to communication protocols and cloud-based analytics. Students will learn the principles of sensor technologies, embedded system design, and IoT networking, along with modern advancements such as edge computing, AI integration, and IoT security. The course emphasizes real-world engineering applications including smart cities, industrial automation, healthcare, and sustainable systems, while fostering critical thinking through AI-assisted learning and research-integrated pedagogy.

UNIT – I	09 Hours

Foundations of Large Language Models:

Fundamentals of language models including their history, evolution, and significance in Natural Language Processing (NLP), word embeddings and neural language models, Transformer architecture covering self-attention mechanism and encoder–decoder structure, encoder layers and encoder stack, and core components of Transformers such as positional encoding, batch normalization, and layer normalization.

(Text Book-1: Chapter 1: 1.1, 1.2, 1.3)
(Text Book-3: Chapter 1: 1.1, 1.2)
(Text Book-4: Chapter 1: 1.1)

UNIT – II	09 Hours

Architectures of Large Language Models:

Decoder-only Large Language Models including GPT architecture, pre-training vs. fine-tuning, and decoding strategies (greedy, beam, top-k, top-p); encoder-only models including BERT architecture,

training objectives, and NLP applications; tokenization techniques such as BPE, WordPiece, and SentencePiece; and an overview of Small Language Models for resource-constrained environments. (Text Book-1: Chapter 2: 2.1, 2.2, 2.3) (Text Book-2: Chapter 3: 3.1, 3.2) (Text Book-4: Chapter 1: 1.1, 1.2)	
UNIT – III	**09 Hours**
Advanced Architectures and Adaptation of LLMs: Encoder–Decoder models including an introduction to models like BART and T5, understanding the text-to-text framework and zero-shot learning; advanced model components covering pre-training strategies, scaling laws, instruction fine-tuning, advanced attention mechanisms, and the Mixture of Experts approach; parameter-efficient fine-tuning (PEFT) techniques for efficient adaptation and inference along with their advantages; and efficient model adaptation through retrieval and tool augmentation, prompt engineering, chains, memory, and agents. (Text Book-1: Chapter 4: 4.1, 4.2) (Text Book-3: Chapter 2: 2.2, 2.3, 2.4) (Text Book-4: Chapter 2: 2.1, 2.2)	
UNIT – IV	**09 Hours**
Introduction to Prompt Engineering Definition, motivation, and importance of prompt engineering, language model control techniques such as temperature and top-p sampling, applications of prompt engineering, elements and structure of prompts including components like instruction, context, input, and output formatting, and prompt formulation strategies for different tasks. (Text Book-1: Chapter 5: 5.1, 5.3) (Text Book-3: Chapter 4: 4.1, 4.2, 4.3) (Text Book-4: Chapter 3: 3.2, 3.3)	
UNIT – V	**09 Hours**
Prompting for Tasks and Advanced Prompt Engineering Techniques: Applications of prompt engineering including summarization, question answering, classification, code generation, and reasoning; role-play and scenario-based prompts; few-shot and zero-shot prompting techniques; advanced methods such as chain-of-thought (CoT), knowledge-augmented prompting, self-consistency, program-aided language models (PAL), and ReAct frameworks; along with hands-on prompt engineering exercises. (Text Book-1: Chapter 7: 7.1, 7.2) (Text Book-3: Chapter 5: 5.1, 5.2, 5.3) (Text Book-4: Chapter 6: 6.1, 6.3)	
• Building Transformer-Based NLP Applications • Building Conversational AI Applications • Building LLM Applications With Prompt Engineering • Rapid Application Development with Large Language Models	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-assisted understanding of language models, word embeddings, and neural LMs for NLP tasks.
- AI-driven implementation of large language models (GPT, BERT), tokenization techniques, and small language model applications.
- AI-enabled design and optimization of encoder–decoder models, advanced architectures, and parameter-efficient fine-tuning techniques.
- AI-guided prompt engineering, including prompt design, control strategies, and task-specific prompt formulation.
- AI-based application of prompting techniques for tasks such as summarization, reasoning, code generation, and advanced frameworks like CoT, PAL, and ReAct.

- Students evaluate limitations, biases, and ethical concerns in language models and NLP systems.
- Students analyze reliability, generalization, and potential biases in large language models and tokenization methods.
- Students assess scalability, efficiency, and trade-offs in advanced architectures and fine-tuning approaches.
- Students examine risks of prompt misuse, bias amplification, and control limitations in prompt engineering.
- Students evaluate trust, interpretability, and ethical implications of advanced prompting techniques and AI-generated outputs.

Applications:

- Language Modeling Systems (Word Embeddings, Neural Language Models, NLP Applications)
- Large Language Models (GPT, BERT, Tokenization Techniques, Small Language Models)
- Advanced NLP Architectures (Encoder–Decoder Models, BART, T5, PEFT, Model Optimization)
- Prompt Engineering Systems (Prompt Design, Control Strategies, Task-specific Prompting)
- AI-driven NLP Applications (Summarization, Question Answering, Code Generation, Reasoning, Advanced Prompting Frameworks)

Mini Project Component

Some sample projects described below simulate the real software engineering problems.

Project 1 – Intelligent Text Summarization System.

- Problem: Develop a system to automatically generate concise summaries from large text documents.
- Concepts used: Transformer models, encoder–decoder architecture (BART/T5), tokenization, fine-tuning.
- Industry relevance: News platforms, research tools, companies like Google, OpenAI.

Project 2 – AI-powered Question Answering System

- Problem: Build a system that can answer user queries based on given context or documents.
- Concepts used: BERT-based models, embeddings, retrieval-augmented generation (RAG), prompt engineering.
- Industry relevance: Chatbots, search engines, virtual assistants (Amazon Alexa, Google Assistant).

Project 3 – Code Generation Assistant

- Problem: Create an AI tool that generates code snippets based on natural language prompts.
- Concepts used: GPT-based models, prompt engineering, decoding strategies, fine-tuning.
- Industry relevance: Developer tools (GitHub Copilot), software companies,

productivity platforms.

CourseOutcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain fundamentals of language models, including word embeddings, neural LMs, and Transformer architecture.	L2
CO2	Apply concepts of large language models such as GPT and BERT, along with tokenization techniques for NLP tasks.	L3
CO3	Design and utilize advanced architectures including encoder–decoder models and parameter-efficient fine-tuning methods.	L3
CO4	Analyze and implement prompt engineering techniques for controlling and optimizing language model outputs.	L3
CO5	Develop AI-driven NLP applications such as summarization, question answering, and code generation using advanced prompting frameworks.	L3

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	2				2		2	2	2
CO2	3	2	2	2	2				2		2	2	2
CO3	3	2	2	2	2				2		2	2	2
CO4	3	2	2	2	2				2		2	2	2
CO5	3	2	2	3	2				2		2	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Speech and Language Processing – Daniel Jurafsky and James H. Martin, 3rd Edition (Draft), Pearson.Comprehensive resource covering fundamentals of NLP, language models, and modern deep learning approaches.
2. Transformers for Natural Language Processing – Denis Rothman, 2nd Edition, Packt, 2021.Focuses on Transformer models, BERT, GPT, and real-world NLP applications.
3. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, and Thomas Wolf, O'Reilly, 2022.Covers Hugging Face ecosystem, fine-tuning, and deployment of transformer-based models.
4. Generative AI with Python and PyTorch – Joseph Babcock et al., Packt, 2023. Introduces generative models, LLMs, and practical implementation using Python.

REFERENCE BOOKS:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016.
2. Language Models for NLP – Sebastian Raschka (Latest Editions / Resources).
3. Hands-On Large Language Models – Jay Alammar and Maarten Grootendorst, O'Reilly, 2024.
4. Prompt Engineering for Generative AI – James Phoenix and Mike Taylor, O'Reilly, 2024.

E-RESOURCES:

1. NPTEL (Courses on NLP, Deep Learning, and AI by IITs)
2. MIT OpenCourseWare (Machine Learning, NLP, and Artificial Intelligence

fundamentals)

3. IEEE research articles on NLP, Large Language Models, and Generative AI

Activity Based Learning (Suggested Activities in Class)

5. Real world problem solving using group discussion. E.g., Designing an NLP-based application such as a chatbot or text summarization system.
6. Role play. E.g., Students act as tokenizer, encoder, decoder, and attention mechanism to simulate Transformer-based data flow.
7. Demonstration of solution to a problem through programming. E.g., Implementing a simple NLP model using Python/Hugging Face for tasks like classification or question answering.
8. Flip class activity. E.g., Prompt engineering techniques, LLM architectures (GPT, BERT), and advanced NLP applications.

****

IoT-Based Data Analytics for Heath Care

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision:
To develop knowledge and skills in IoT-enabled smart healthcare systems for designing secure, intelligent, and sustainable health monitoring and analytics solutions using AI, data analytics, and blockchain technologies.

Prerequisites:

Course Learning Objectives:

This Course will enable students to:

1. Understand IoT architecture and standards in healthcare environments.
2. Establish secure data collection and transmission models.
3. Apply AI/ML techniques for health data analytics.
4. Integrate blockchain for data integrity and sharing.
5. Develop sustainable smart health analytics solutions..

Broad Course Description

This course integrates Internet of Things concepts with secure data analytics for healthcare applications. It covers sensor networks, IoT communication protocols, data acquisition pipelines, AI-enabled health analytics, and privacy-preserving data sharing using blockchain, enabling students to design sustainable and secure smart healthcare solutions.

UNIT – I	**06 Hours**

IoT ARCHITECTURE AND HEALTH DEVICES:
Sensing technologies (BP, HR, ECG). IoT stack overview – device, gateway, cloud, application. Communication protocols (MQTT, CoAP, BLE, LoRa). Data formats (JSON, CBOR). Healthcare standards – HL7, FHIR.
(Text Book 1 : Ch. 1 1.1–1.6; Text Book 2 : Ch. 2 2.1–2.3)

UNIT – II	**06 Hours**

IoT NETWORKING AND SECURITY:
End-to-end communication models. Security challenges in IoT and M2M systems. Device authentication. Cloud and edge security. Secure IoT middleware and firmware updates
(Text Book 1 : Ch. 3 3.1–3.5; Text Book 2 : Ch. 4 4.1–4.4)

UNIT – III	**06 Hours**

DATA ACQUISITION AND STORAGE:
Streaming data pipelines (MQTT, Kafka). Time-series databases (InfluxDB, MongoDB). Data cleaning, synchronization, missing value imputation. Data privacy – anonymization and differential privacy.
(Text Book 1 : Ch. 5 5.1–5.5;
Text Book 2 : Ch. 6 6.1–6.3)

UNIT – IV	**06 Hours**

AI-BASED HEALTH DATA ANALYTICS:
Predictive analytics models (SVM, Random Forest, CNN for ECG). Federated learning for distributed data models. Early warning systems and real-time visual dashboards. Performance evaluation metrics (precision, recall, ROC curve). ML algorithms for sensor data.

(Text Book 2 : Ch. 8 8.1–8.5)	
UNIT – V	**06 Hours**
BLOCKCHAIN INTERFACING AND ETHICAL USE: Smart contracts for health data access control. Blockchain data storage formats and timestamping. Audit trails and data verification process. GDPR and HIPAA compliance integration. (Text Book 1 : Ch. 7 7.1–7.4; Text Book 2 : Ch. 9 9.2–9.4)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

1. Edge AI for low-power devices.
2. Responsible AI and Bias Mitigation in Clinical Analytics.

Applications:

3. Monitoring patient vitals using connected health sensors.
4. Detecting medical anomalies through AI-based analytics on IoT data.
5. Storing and processing patient data securely on cloud and edge platforms.
6. Integrating blockchain for tamper-proof health records.
7. Building predictive healthcare dashboards for early diagnosis.

Mini Project Component

The following can be considered as the project-based activities in the course:

1. **Project:** Connectivity simulation of multiple sensor nodes (Raspberry Pi / Arduino).
2. **Mini Lab:** Implement token-based device authentication.
3. **Activity:** Design a cloud data pipeline using AWS IoT Core or ThingSpeak.
4. **Mini Project:** Anomaly detection in vital sign datasets.
5. **Capstone:** Prototype “patient-data ledger” with integrity check.

Course Outcome	Description	Bloom’s Taxonomy Level
At the end of the course the student will be able to:		
CO1	Describe IoT architecture and standards for health systems.	L2
CO2	Implement secure IoT networks using cryptographic techniques.	L3
CO3	Develop data pipelines for health analytics.	L3
CO4	Apply AI models to analyze sensor data.	L3
CO5	Integrate blockchain for privacy-preserving computation.	L4

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			2						2	2	
CO2	2	3	3		3			2			2	2	
CO3	2	3	3	2	3						2	3	
CO4	2	3	3	2	3						2	3	
CO5	2	2	3		3			2			2	3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti – IoT: A Hands-On Approach, Universities Press.
2. Dietmar Putz – AI for IoT Applications, Springer.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Edge AI and Jetson for IoT	III, IV	Edge ML deployments
Deep Learning for Anomaly Detection	IV	Predictive Health analytics

E-RESOURCES:

1. nptel.ac.in
2. coursera.org
3. aws.amazon.com
4. developer.nvidia.com
5. cse29-iiith.vlabs.ac.in

Activity-Based Learning (Suggested Activities in Class)

1. Connect a sensor to an IoT platform (AWS IoT Core / ThingSpeak).
2. Record and visualize temperature or heart-rate data streams.
3. Perform data cleaning on real IoT sensor files using Python.
4. Apply a simple ML algorithm for anomaly detection in vitals.
5. Present a use case on secure IoT ecosystem in healthcare.

****

Game Development using Unity

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	04 Hours	**Total Hours**	:	30HrsL +15Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: To develop creative and technical expertise in game development using Unity for designing interactive, immersive, and intelligent gaming experiences across multiple platforms using modern game design and programming techniques.

Course Learning Objectives:

This Course will enable students to:

1. Develop proficiency in Unity for creating 2D and 3D games.
2. Learn C# scripting for game logic, player controls, and game systems.
3. Implement physics, animation, AI behaviors, and audio in Unity.
4. Build and deploy complete games for desktop and mobile platforms.

Broad Course Description

Game Development using Unity is a project-driven course covering Unity editor fundamentals, C# scripting (MonoBehaviour lifecycle), physics, 2D Tilemap, animation (Animator/Blend Trees/Cinemachine), NavMesh AI, Unity UI (uGUI/TextMeshPro), audio management, save systems, Shader Graph, particle systems (Niagara-style), optimization, and deployment to Android/iOS/PC.

UNIT – I	**06 Hours**

Unity Foundations & C# Scripting: Unity Editor – Hierarchy, Inspector, Project, Scene views, GameObjects, Components, Prefabs, MonoBehaviour lifecycle, Input System, Instantiate/Destroy, Tags/Layers, Scene management.

• Harrison Ferrone, "Learning C# by Developing Games with Unity" 7th Ed Ch. 1–5 (Packt, 2023)
• Unity Learn Official Curriculum (learn.unity.com)

UNIT – II	**06 Hours**

Physics, Collision & 2D Game Development: Rigidbody and Collider – 3D and 2D, Physics materials, Collision detection events, Layer-based collision, 2D workflow – Sprites, Sprite Atlas, Tilemap, 2D joints, Coroutines, Object pooling.

• Harrison Ferrone Ch. 7–10
• John P. Doran, "Unity 2022 by Example" Ch. 2–4 (Packt, 2023)

UNIT – III	**06 Hours**

Animation & Character Systems: Animation clips, Animator Controller, Transitions, Parameters, Blend Trees, Root motion vs. in-place animation, Cinemachine, Character Controller vs. Rigidbody, Ragdoll physics, IK basics, Particle System.

• John P. Doran Ch. 5–7
• Unity Animation Documentation (docs.unity3d.com/Manual/AnimationSection.html)

UNIT – IV	**06 Hours**

AI, UI & Game Systems: NavMesh AI – patrol, chase, attack state machines, Scriptable Objects, Unity uGUI – Canvas, Button, Text, Image, Slider, Health bars, score HUD, TextMeshPro, Audio Source/Mixer, Save system – PlayerPrefs/JSON, Game Manager/Singleton.

• Harrison Ferrone Ch. 11–14
• John P. Doran Ch. 8–10

UNIT – V	06 Hours
Lighting, Shaders, Optimization & Deployment: Unity lighting – Baked vs. Real-time, Light Probes, Reflection Probes, Shader Graph, Post-processing, Performance optimization – batching, LOD, Profiler, Mobile build settings, Android/iOS, Unity Asset Store, Publishing to Google Play/App Store/itch.io. • John P. Doran Ch. 11–13 • Unity Performance Best Practices (docs.unity3d.com/Manual/BestPracticeGuides.html)	

NVIDIA Deep Learning Institute (DLI) Course Mapping

The following NVIDIA DLI courses are recommended to supplement the subject. Free courses are accessible at learn.nvidia.com. Paid instructor-led workshops (ILT) are available via NVIDIA partner institutions.

UNIT	NVIDIA DLI Course Name	Type	Duration	URL / Code
Units 3–4	Getting Started With Deep Learning Foundational deep learning – prerequisite for Unity ML-Agents integration and AI NPC development covered in Unit 4.	Self-Paced	2 Hrs	FREE DLI+C-FX-01+V1
Unit 4	Getting Started With AI on Jetson Nano Practical AI deployment on edge hardware – models trained here can be integrated as Unity NPC AI via ONNX export.	Workshop (ILT)	8 Hrs	PAID DLI-ILT-GAIJN
Units 3–5	Build Beautiful, Custom UI for 3D Tools on NVIDIA Omniverse Self-paced UI design for 3D platforms – directly applicable to Unity UGUI/HUD design principles and 3D tool UI patterns.	Self-Paced	2 Hrs	FREE DLI+S-OV-12+V1
Units 4–5	Generative AI Explained GenAI overview – applicable to AI-generated game assets, procedural content, and ML-Agents-based NPC behavior.	Self-Paced	1 Hr	FREE DLI+S-FX-07+V1
Unit 5	Bootstrapping Computer Vision Models with Synthetic Data ILT: use NVIDIA Omniverse to generate synthetic training data for computer vision – applicable to AI enemy vision systems and game perception.	Workshop (ILT)	8 Hrs	PAID DLI-ILT-BCVMSD
Unit 5	Synthetic Data Generation for Training Computer Vision Models Self-paced: generating synthetic game-world data for training vision-based AI models – advanced Unity AI integration topic.	Self-Paced	2 Hrs	FREE DLI-SP-SDGCVM

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-Assisted Code Generation: Use GitHub Copilot in Unity to generate C# scripts.
- Procedural Content with AI: Implement procedural level generation.
- Behavior AI Reflection: Compare state machine AI with ML-based agents.
- Automated Playtesting: Explore Unity ML-Agents for automated testing.
- AI Art Pipeline: Use AI-generated 2D/3D assets and discuss IP and quality.

Applications

- Mobile and PC Game Publishing

- Serious Games for Healthcare and Education
- Architectural VR Walkthroughs
- Interactive Training Simulations
- AR Applications for Marketing

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging. Lab Experiments

- Build 2D platformer with player movement, jump, and obstacle collision.
- Create Tilemap-based level with rule tiles and scrolling camera.
- Implement Animator Controller with 4 states and blend tree transitions.
- Design 3D third-person controller with Cinemachine.
- Build NavMesh-based enemy AI with patrol, detection, chase.
- Create complete HUD with health bar, score, timer, and pause menu.
- Implement save/load system using JSON serialization.
- Build particle effect system for visual feedback.
- Implement mobile touch controls and build for Android/iOS.
- Submit fully playable original game with win/lose conditions and sound.

Mini Project Component

Industry-driven projects that simulate real software engineering problems. Project 1 – 2D Mobile Platformer

Problem: 2D platformer with 3 levels, enemies, collectibles, score, mobile touch controls.

Concepts used: Unity 2D, Tilemap, Animation, Physics, Mobile deployment

Industry relevance: Mobile game publishing on Google Play / App Store.

Project 2 – 3D Top-Down Shooter

Problem: 3D shooter with AI enemies, weapon systems, power-ups, and progression. Concepts used: 3D Unity, NavMesh AI, Scriptable Objects, UI

Industry relevance: Action game genre – widely played across platforms.

Project 3 – Puzzle Adventure Game

Problem: 3D puzzle-adventure with physics puzzles, inventory system, and narrative.

Concepts used: Unity Physics, Game Manager, Save system,

Lighting Industry relevance: Indie puzzle games on Steam and itch.io.

Course Outcomes

CO	Description	Bloom's
At the end of the course the student will be able to:		
CO1	Navigate Unity editor proficiently and write C# scripts for game logic, input handling, and scene management.	L3
CO2	Implement 2D/3D physics simulations and collision detection; create complete 2D games using Tilemap workflows.	L3
CO3	Design character animation systems using Animator Controller, Blend Trees, and Cinemachine.	L3
CO4	Build complete game systems: AI pathfinding, UI/HUD, audio management, and persistent save systems.	L4
CO5	Optimize, polish, and deploy a complete original game to target platforms.	L5

Table: Mapping Levels of COs to POs / PSOs													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	3	3	3						2	3	3
CO2	3	3	2	3	3						2	3	2
CO3	3	3	3	3	2						2	3	2
CO4	3	3	3	3	3						3	3	3
CO5	3	3	3	3	3						3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools

- Unity 2022 LTS / 2023
- Visual Studio / Rider
- Cinemachine
- TextMeshPro
- Unity ML-Agents Toolkit
- Blender

TEXT BOOKS:

1. John P. Doran, "Unity 2022 by Example", Packt Publishing, 2023.
2. Jason Gregory, "Game Engine Architecture", 3rd Ed, CRC Press, 2018.
3. Harrison Ferrone, "Learning C# by Developing Games with Unity", 7th Ed, Packt, 2023.

REFERENCE BOOKS:

1. Unity Technologies, Official Unity Manual and Scripting Reference (docs.unity3d.com).
2. Adam Thorn, "The Unity Game Developer's Cookbook", Addison-Wesley, 2013.

E-RESOURCES:

- Unity Learn (learn.unity.com)
- Brackeys YouTube Channel
- Game Dev TV (gamedev.tv)
- NVIDIA DLI: learn.nvidia.com

Activity Based Learning (Suggested Activities in Class)

- Monthly game jams using Unity (48-hour events).
- Portfolio reviews – peer and instructor critique of game builds.
- Publish games to itch.io for community feedback.

****

Quantum Machine Learning

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30hrs L + 15hrs J
L–T–P-J	:	2–0–0-2			

Prerequisites: Linear Algebra, Probability, Machine Learning

Course Learning Objectives:

This Course will enable students to:

1. Understand fundamental concepts of machine learning and their quantum extensions.
2. Learn techniques for encoding classical data into quantum states.
3. Analyze variational quantum circuits as machine learning models.
4. Explore quantum kernel methods and quantum optimization techniques.
5. Develop applications using Quantum Machine Learning frameworks.

Course Vision:

To empower students with the ability to integrate quantum computing principles with machine learning techniques for solving complex data-driven problems. To cultivate innovation and ethical responsibility in applying quantum-enhanced algorithms to real-world engineering, scientific, and societal challenges.

Broad Course Description

The course Quantum Machine Learning provides a comprehensive understanding of integrating machine learning techniques with quantum computing principles to develop advanced intelligent systems. It covers fundamental concepts of machine learning, followed by quantum data representation methods such as basis and amplitude encoding. The course explores variational quantum circuits and quantum neural networks as learning models, along with quantum kernel methods and optimization using Ising models. Through real-world case studies, students gain practical insights into applications in healthcare, finance, and image processing, equipping them to design and analyze hybrid quantum–classical machine learning solutions.

UNIT – I	08 Hours

Introduction to Machine Learning:

Background, The Toy Example of Quantum Algorithm for classification, Examples of Typical machine learning problems, The Three Ingredients of a Learning Problem, Risk Minimisation in Supervised Learning, Training in Unsupervised Learning, Methods in Machine Learning.

(Text Book-1: Chapter 1, 2)

UNIT – II	10 Hours

Representing Data on a Quantum Computer:

Encoding Binary Inputs into Basis States, Arbitrary State Preparation for Amplitude Encoding, Encoding Inputs as Time Evolutions, Encoding a Dataset via the Hamiltonian, Data Encoding as a Feature Map.

(Text Book-1: Chapter 4)

UNIT – III	09 Hours
Variational Circuits as Machine Learning Models: How to Interpret a Quantum Circuit as a Model, Which Functions Do Variational Quantum Models Express? Training Variational Quantum Models, Quantum Circuits and Neural Networks. (Text Book 1: Chapter 5)	
UNIT – IV	**09 Hours**
Quantum Models as Kernel Methods and Ising Model: The Connection Between Quantum Models and Kernel, Quantum Computing, Feature Maps and Kernels, Examples of Quantum Kernels, The RKHS of Quantum Kernels, Kernel-Based Training, Quantum Extensions of Ising Models, Quantum Annealing. (Text Book1: Chapter 6: 6.1, 6.2, 6.3, 6.4, 6.5. Chapter 8: 8.1, 8.2)	
UNIT – V	**09 Hours**
Case Studies and Applications of QML: Healthcare Diagnosis using QML, Financial Market Prediction, Image Classification using Quantum Feature Maps, Image Classification using Quantum Feature Maps	

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **Quantum-Enhanced Machine Learning:** Integrate classical machine learning concepts with quantum computing principles to design hybrid models that improve learning efficiency and scalability.
- **Advanced Data Representation:** Utilize quantum data encoding techniques (basis states, amplitude encoding, feature maps) to represent complex datasets in high-dimensional spaces for improved model performance.
- **Variational Learning Models:** Apply Variational Quantum Circuits (VQC) and Quantum Neural Networks (QNN) as trainable models, combining quantum circuits with classical optimization techniques.
- **Kernel-Based Quantum Intelligence:** Leverage quantum kernels and feature maps to enhance classification and pattern recognition tasks, enabling more expressive learning models.
- **Optimization and Decision Systems:** Use quantum annealing and Ising models to solve complex optimization problems in AI applications efficiently.
- **Responsible & Explainable Quantum AI:** Ensure fairness, transparency, and interpretability in quantum machine learning models while addressing challenges such as bias, uncertainty, and reproducibility.

Suggested Tools:

- Tensorflow Quantum
- PennyLane
- Google Colab
- IBM Quantum Expereince
- Qiskit Machine Learning

Applications:

- Healthcare diagnosis and disease prediction using quantum ML models
- Financial market prediction and risk analysis
- Image classification and pattern recognition using quantum feature maps
- Optimization problems in logistics and resource allocation
- Fraud detection and anomaly detection systems
- Drug discovery and molecular data analysis
- Smart decision systems using hybrid quantum–classical AI

Mini Project Component

Industry-Driven Projects (IBM / Google Quantum AI Style)

These projects simulate real-world **quantum machine learning and intelligent system applications**.

Project 1 – Quantum Classification Model

- **Problem:** Develop a classification model using Variational Quantum Circuits (VQC)
- **Concepts used:** Feature maps, parameterized circuits, training
- **Industry relevance:** Healthcare, finance, predictive analytics

Project 2 – Case Study-Based QML Application

- **Problem:** Implement a real-world case study (healthcare/finance/image classification)
- **Concepts used:** QML pipeline, hybrid models, performance evaluation
- **Industry relevance**: Industry-ready AI solutions

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain machine learning fundamentals and quantum learning concepts	L2
2	Apply data encoding techniques for representing classical data in quantum systems	L3
3	Analyze variational quantum circuits as machine learning models	L4
4	Evaluate quantum kernel methods and quantum optimization approaches.	L3
5	Evaluate quantum kernel methods and quantum optimization approaches.	L6

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	1	1	1							2	
CO2	3	3	2	2	3							3	2
CO3	2	3	2	3	2							3	2
CO4	2	3	2	3	2							3	2
CO5	3	3	3	3	3						2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Maria Schuld · Francesco Petruccione, "**Machine Learning with Quantum Computers",** Second Edition, Springer.

REFERENCE BOOKS:

1. Maria Schuld and Francesco Petruccione – *Supervised Learning with Quantum Computers*
2. Peter Wittek – *Quantum Machine Learning: What Quantum Computing Means to Data Mining*
3. Michael Nielsen and Isaac Chuang – *Quantum Computation and Quantum Information*

E-Resources:

- https://qiskit.org
- https://pennylane.ai
- https://quantum.ibm.com
- https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Quantum data encoding experiment
Activity 2: Quantum circuit based classifier implementation
Activity 3: Hybrid quantum–classical ML model design
Activity 4: Mini project on quantum machine learning application

****

Business to Business Marketing

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:	**Credits:**	03
Hours / Week	: 03 Hours	**Total Hours**:	45 Hrs L
L–T–P–J	: 3–0–0-0		

Course Vision:

The course aims to develop a strong foundation in Business-to-Business (B2B) marketing concepts, strategies, and relationship management for understanding modern industrial and organizational markets. It enables students to analyze organizational buying behaviour, market segmentation, pricing, and channel management strategies for creating business value and competitive advantage. The course also equips learners with communication, decision-making, and data-driven marketing skills required to manage products, services, customer relationships, and digital marketing practices in complex B2B environments.

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamentals of B2B marketing and the structure of business markets.
2. To analyze organizational buying behaviour and decision-making processes.
3. To apply strategic marketing concepts for managing products, services, and business portfolios.
4. To evaluate segmentation, targeting, positioning, and pricing strategies in B2B markets.
5. To develop skills in business communication, channel management, and relationship marketing.

Broad Course Description

This course provides a comprehensive understanding of Business-to-Business (B2B) marketing, focusing on how organizations create value and build relationships in industrial and institutional markets. It covers the characteristics of business markets, organizational buying behaviour, and strategic marketing processes, followed by product, service, and channel management in B2B contexts. The course also explores segmentation, targeting, positioning, pricing strategies, and relationship marketing, along with modern digital marketing practices. Emphasis is given to business communication, sales force management, and logistics to help students develop practical skills for managing complex B2B environments and making data-driven marketing decisions.

UNIT – I	**09 Hours**

Introduction to B2B Marketing: Business marketing and Business market customers, Market structure, Environment and Characteristics of Business Marketing, Strategic role of marketing, Commercial enterprises, Commercial and institutional customers.

UNIT – II	**09 Hours**

Organizational Buying and Buyer Behaviour: Organizational buyers' decision process - A Stepwise Model and A Process Flow Model, Organizational and business markets - Government as a customer - Commercial enterprises - Commercial and institutional Customers.

UNIT – III	**09 Hours**

B2B Marketing Strategy: Strategy making and strategy management process, Industrial product strategy–Managing Products for Business Markets-Managing Services for Business Markets-Managing Business
Market Channels the Growth-Share Matrix, Multifactor Portfolio Matrix, The Balanced Scorecard.

UNIT – IV	09 Hours
B2B Marketing STP: Market Segmentation, basic framework of segmentation, choosing target segments and positioning-Pricing strategies for Business Markets, B2B Advertising, Competitive bidding, Relationship marketing and CRM.	
UNIT – V	**09 Hours**
Business Marketing Communications and Channels- B2B Advertising, Digital marketing, - Trade shows, exhibitions, business meets - Managing the sales force - Deployment analysis -Business marketing channels and participants - Channel design and management decisions -B2B logistics management.	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-driven Market Intelligence:**
  Use AI tools to analyze B2B market trends, customer needs, and competitor strategies while ensuring ethical data usage and unbiased insights.
- **Intelligent Customer Relationship Management:**
  Apply AI in CRM systems to predict customer behaviour, personalize interactions, and enhance long-term business relationships with transparency.
- **AI-enabled Strategic Decision Making:**
  Utilize AI-based analytics for pricing, segmentation, and channel decisions with explainable and data-driven approaches.
- **Smart Sales and Marketing Automation:**
  Implement AI in digital marketing campaigns, lead generation, and sales forecasting while maintaining privacy and fairness.
- **Responsible AI in Business Marketing:**
  Evaluate ethical, legal, and societal implications of AI in B2B marketing, ensuring trust, accountability, and sustainable practices.

Suggested Tools:

- HubSpot
- SEMrush
- Mailchimp
- Miro
- Notion

Applications:

- Market segmentation and targeting in industrial markets.
- Customer relationship management and key account handling.
- Supply chain and distribution channel optimization.
- Digital marketing and lead generation in B2B platforms.
- Pricing and competitive bidding strategies.
- Sales force automation and performance analysis.

Project Component

- Analyze a selected industry to identify potential B2B opportunities, customer segments, and competitors.
- Design an effective distribution channel and logistics plan for a B2B product.
- Develop a LinkedIn/email-based marketing campaign for a business product or service.
- Conduct a case study on how a company makes purchasing decisions and map the decision-making process.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Understand the concepts, structure, and environment of B2B marketing.	L2
CO2	Analyze organizational buying behavior and decision-making models in business markets.	L4
CO3	Apply B2B marketing strategies for product, service, and portfolio management.	L3
CO4	Evaluate STP strategies, pricing, and relationship marketing approaches in B2B contexts.	L5
CO5	Design effective B2B communication, sales force, and channel management strategies.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	2	2	3			2	2	2	2		3
CO2	2	2	2	2	3			2	2	2	2		3
CO3	2	2	2	2	3			2	2	2	2		3
CO4	2	2	2	3	3			2	2	2	2		3
CO5	2	2	2	2	3			2	2	2	2		3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Krishna K. Havaldar, Vasant M. Cavale, *Business Marketing: Text and Cases*, 4th Edition, McGraw Hill Education, 2017.

REFERENCE BOOKS:

1. Michael D. Hutt, Dheeraj Sharma, Thomas W. Speh, B2B Marketing: A South Asian Perspective Cengage 2014, 11th ed.
2. Sharad Sarin, Business Marketing: Concepts and Cases McGraw Hill 2013, 1st ed.
3. James C. Anderson, Das Narayandas, James A. Narus and D.V.R. Seshadri, Business Market Management (B2B): Understanding, Creating, and Delivering Value, Pearson 2010, 3rd ed.

4. Robert Vitale, WaldemarPfoertsch, Joseph Giglierano, Business to Business, Marketing, Pearson 2011
5. Krishna K Havaldar, Business Marketing: Text and Cases McGraw Hill 2014, 4th ed.
6. Armstrong, Gary and Philip Kotler, "Principles of Marketing", Prentice Hall, New Delhi, 2006

E-RESOURCES:

- https://www.coursera.org/learn/b2b-marketing
- https://onlinecourses.nptel.ac.in/noc23_mg120/preview
- https://www.b2bmarketing.net/resources/
- https://www.forrester.com/b2b-marketing/content-strategy-resources/

Activity Based Learning (Suggested Activities in Class)

- Analyze market structure, customers, and competitive environment, followed by a short presentation.
- Conduct a role-play where students simulate the organizational buying process (buyer, influencer, decision-maker) to understand decision-making dynamics.
- Design a sales force structure and CRM strategy for managing key business clients.

****

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Introduction To Internet Of Things, IIT Kharagpur [NPTEL] 12 WEEKS [Introduction To Internet Of Things | SWAYAM](https://onlinecourses.nptel.ac.in/noc26_cs37) https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs37

Note: CID = NPTEL Course ID

Generative AI with Diffusion Models

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P-J	:	3–0–0-0			

Course Vision:

The course aims to develop strong foundations in Generative AI, diffusion models, and multimodal intelligence for building advanced AI-driven applications. It enables students to understand and apply modern techniques in text generation, image synthesis, prompt engineering, and vision-language modelling using scalable AI architectures and APIs. The course also equips learners with practical skills to design ethical, secure, and responsible AI systems capable of generating and integrating multimodal content for real-world applications across diverse domains.

Prerequisites: Basic Programming, Mathematics

Course Learning Objectives:

This Course will enable students to:

2. Understand the principles of generative AI, including text, image, and multimodal models.
3. Apply prompt engineering and fine-tuning techniques for domain-specific text generation tasks.
4. Analyze diffusion and GAN-based models for image synthesis and enhancement.
5. Design and deploy generative AI systems using APIs and scalable architectures.
6. Evaluate ethical, bias, and safety issues in real-world AI deployments.

Broad Course Description

This course provides an in-depth understanding of Generative AI, Diffusion Models, and Vision-Language Models, focusing on both theoretical foundations and practical implementation. Students will explore advanced techniques in text generation, image synthesis, and multimodal learning, along with deployment strategies for real-world applications. The course emphasizes hands-on development, ethical considerations, and scalable AI system design, enabling learners to build intelligent systems capable of generating, interpreting, and integrating multimodal data across domains.

UNIT – I	09 Hours

ADVANCED TEXT GENERATION AND CUSTOMIZATION

Fine-tuning and customizing large language models for specific domains, Prompt engineering and chaining prompts for complex tasks, Case studies: automated report generation, creative writing, summarization. (Text Book-1)

UNIT – II	08 Hours

IMAGE & MULTIMEDIA SYNTHESIS

Using GANs and diffusion models for image creation and editing, Tools for rapid image prototyping and style transfer, Multimedia content generation: audio, video, and cross-modal use cases.
(Text Book1)

UNIT – III	08 Hours

DEPLOYMENT AND INTEGRATION

Incorporating generative AI into web and mobile applications (e.g., APIs, plugins),Ethics, safety, and bias management in deployment, Scalability and performance considerations.
Text Book1)

UNIT – IV	10 Hours
DIFFUSION MODELS Introduction, Denoising Diffusion Probabilistic Models (DDPMs), Classifier and Classifier-Free Diffusion Guidance, Text-conditioned Diffusion Models: Sampling, Prediction Space, Noise Schedules, Architectures. (Text Book1)	
UNIT – V	**10 Hours**
VISION-LANGUAGE MODELS Self-Supervised Learning: Introduction, Simple Framework for Contrastive Learning of Visual Representation (SimCLR), Contrastive Learning, Vision-Language Models, Contrastive Language-Image Pre-training (CLIP). (Text Book1)	
● Generative AI Explained ● Introduction to Deploying RAG Pipelines for Production at Scale ● Deploying and Optimizing AI Inference at Scale ● Deploying RAG Pipelines for Production at Scale	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- This course integrates AI as both a **learning tool and subject of study**. Students will:
- Use AI tools for **code generation, prototyping, and experimentation**
- Critically reflect on **AI-generated outputs vs human intelligence**
- Analyze **bias, hallucination, and ethical risks**
- Develop **authentic solutions**, ensuring:
- Original problem formulation
- Human-in-the-loop validation
- Transparent AI usage documentation
- How AI assisted vs how I improved the solution
- Limitations of AI outputs in my project
- Ethical considerations in my design

AI Tools:

Programming: Python

Libraries: PyTorch, TensorFlow, Hugging Face

Transformers Generative AI Tools: ChatGPT, DALL·E,

Stable Diffusion Vision Tools: OpenCV, CLIP models

Deployment: FastAPI, Flask, Streamlit

Applications:

- Automated report and content generation
- AI-powered chatbots and assistants
- Image generation and editing systems
- Medical image synthesis and analysis
- Cross-modal search engines (text ↔ image)
- Video/audio content generation
- Creative AI (art, storytelling, design)

Mini Project Component

- Some sample projects described below simulate the real software engineering problems.

1. Domain-Specific AI Report Generator
   - Fine-tuned LLM for generating technical/research reports
2. Text-to-Image Generator using Diffusion Models
   - Prompt-based image synthesis with style control
3. AI-Powered Multimodal Search Engine
   - Retrieve images using text queries (CLIP-based)
4. Smart Content Creation Assistant
   - Generates blogs, summaries, and creative content
5. Ethical AI Analyzer Tool
   - Detects bias and unsafe outputs in generative AI systems

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Explain concepts of generative AI, GANs, diffusion models, and vision-language models	L3
CO2	Apply prompt engineering and fine-tuning techniques for text generation	L3
CO3	Analyze and implement image synthesis models using GANs and diffusion	L4
CO4	Design and deploy AI-powered applications integrating multimodal models	L6
CO5	Evaluate ethical, bias, and scalability challenges in AI systems	L4

COS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	-	-	-	-	-	-	-	-	3	2
CO2	2	3	2	1	2	-	-	-	1	-	-	3	2
CO3	2	3	3	2	2	-	-	-	1	-	-	3	2
CO4	2	2	3	3	3	1	-	-	2	2	2	3	2
CO5	2	3	2	2	1	2	1	3	2	2	1	3	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Textbooks:

1. Hands-On Generative AI with Transformers and Diffusion Models, Omar Sanseviero et al., O'Reilly Media, 1st Edition, 2024
2. Applied Generative AI for Beginners, Akshay Kulkarni et al., Apress, 1st Edition, 2023.

REFERENCE BOOKS:

1. Image Generation Models, Vladimir Bok, Manning Publications, 1st Edition, 2025
2. Master Diffusion Models: A Practical Guide to Generative AI, Ashley, Independently Published, 1st Edition, 2025.

E resources:

1. Coursera: Best Generative AI Courses & Certificates
2. Udemy: Top Generative AI (GenAI) Courses Online

Activity Based Learning (Suggested Activities in Class)

- Prompt Engineering and Text Generation Lab
- Text-to-Image Generation using Diffusion Models
- Generative AI Application Deployment Project
- Diffusion Model Visualization and Analysis
- Vision-Language Model (CLIP) Implementation

****

Blockchain Technology

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop a strong foundation in blockchain technology, smart contracts, and decentralized systems for building secure and transparent digital platforms. It enables students to understand blockchain architecture, consensus mechanisms, and the integration of blockchain with IoT and AI technologies for modern applications. The course also promotes awareness of security, governance, ethical practices, and sustainable blockchain solutions required for next-generation trusted digital ecosystems.

Course Learning Objectives:

This Course will enable students to:

1. Understand blockchain architecture and consensus mechanisms.
2. Write and deploy smart contracts for secure applications.
3. Integrate IoT devices and data into blockchain systems.
4. Analyze blockchain performance, governance, and ethics.
5. Explore AI synergy and green blockchain designs.

Broad Course Description

This course provides a comprehensive understanding of blockchain architecture, distributed ledgers, and consensus mechanisms. It focuses on smart contract design, decentralized applications, and blockchain integration with IoT and AI systems, highlighting transparency, security, and ethical aspects crucial for next-generation digital trust platforms.

UNIT – I	06 Hours

BLOCKCHAIN PRINCIPLES:

Distributed ledger technologies. Blockchain structure, transactions, blocks, hashes. Cryptographic primitives. Peer-to-peer communication and consensus mechanisms. Types of blockchain (public, private, consortium).

(Text Book 1 : Ch. 1 1.1–1.4)

UNIT – II	06 Hours

CONSENSUS ALGORITHMS:

PoW, PoS, DPoS, PBFT, Raft. Mining concept and difficulty adjustment. Energy consumption vs efficiency metrics. Smart contract execution model.

(Text Book 1 : Ch. 3 3.1–3.5; Text Book 2 : Ch. 2 2.3–2.5)

UNIT – III	06 Hours

SMART CONTRACTS AND DApps :

Ethereum architecture and EVM. Solidity language basics. Smart contract debugging and deployment (Truffle/Ganache). Token creation (ERC-20 / NFT).

Text Book 1 : Ch. 5 5.1–5.4)

UNIT – IV	06 Hours

BLOCKCHAIN INTEGRATION:

Blockchain in Healthcare, Supply Chain, Finance. Interoperability and sidechains (Polkadot, Cosmos). Identity and Access Management on blockchain. AI–Blockchain convergence (federated learning audit

trail). (Text Book 2 : Ch. 4 4.1–4.4)	
UNIT – V	**06 Hours**
PERFORMANCE, ETHICS & EMERGING TRENDS: Scalability issues and Layer-2 solutions (Lightning, Plasma). Quantum-resistant blockchains. NFTs, DeFi, and Web3. Sustainability and energy-efficient mining. (Text Book 1 : Ch. 7 7.1–7.3; Text Book 2 : Ch. 5 5.1–5.2)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

1. AI driven fraud detection on blockchains.
2. Reinforcement learning for consensus optimization.

Applications:

1. Create and deploy a basic smart contract using Remix/Metamask.
2. Build a simple blockchain prototype in Python to log transactions.
3. Trace and visualize block hash links to explain immutability.
4. Conduct a group discussion on public vs private blockchains.
5. Demonstrate a demo use case of IoT Blockchain integration.

Mini Project Component

The following project-based activities can be conducted in the course:

1. Journal: Draw a transaction lifecycle diagram.
2. Project: Implement PoW simulation in Python.
3. Activity: Case study – IBM Food Trust or Hyperledger Fabric.
4. Capstone: Prototype green blockchain ledger with energy usage estimation.
5. Project: Design a consent management smart contract for IoT data.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain the fundamental concepts, architecture, and working of blockchain systems along with cryptographic foundations.	L2
CO2	Analyze and apply various consensus mechanisms such as Proof of Work, Proof of Stake, and PBFT for secure distributed agreement.	L3
CO3	Design and implement smart contracts and decentralized applications (DApps) using suitable development frameworks.	L3
CO4	Integrate blockchain with IoT and AI systems for secure data management and provenance tracking.	L4
CO5	Evaluate blockchain performance, scalability, and ethical implications to propose sustainable, real-world solutions.	L4

Table: Mapping Levels of COs to POs / PSOs		
COS	**Program Outcomes (POs)**	**Program Specific Outcomes (PSOs)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			2						2	2	
CO2	2	3	2	2	2						2	3	
CO3	2	2	3	2	3						2	3	
CO4	2	2	3	2	3			2			2	3	
CO5	2	3	3	2	2		2	2			2	3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Imran Bashir – Mastering Blockchain, 3e, Packt.
2. Arvind Narasimha – Blockchain for Enterprise Applications, CRC Press.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Accelerated Computing for Blockchain Analytics	II–V	GPU-enhanced validation
AI for Fraud Detection	IV–V	Smart contract monitoring

E-RESOURCES:

1. ptel.ac.in
2. coursera.org
3. edx.org
4. ethereum.org
5. developer.ibm.com

Activity-Based Learning (Suggested Activities in Class)

1. Create and deploy a basic smart contract using Remix/Metamask.
2. Build a simple blockchain prototype in Python to log transactions.
3. Trace and visualize block hash links to explain immutability.
4. Conduct a group discussion on public vs private blockchains.
5. Demonstrate a demo use-case of IoT-Blockchain integration.

****

AR/VR & Game-Theoretic

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L+15Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision:

The course aims to develop knowledge and practical skills in Augmented Reality (AR), Virtual Reality (VR), and Algorithmic Game Theory for designing immersive and intelligent interactive systems. It enables students to create AR/VR applications using modern development platforms and understand strategic decision-making concepts such as Nash Equilibrium, auctions, and mechanism design. The course also prepares students to apply game-theoretic principles and immersive technologies in areas such as gaming, simulation, multi-agent systems, human-computer interaction, and intelligent digital environments.

Course Learning Objectives:

This Course will enable students to:

1. Understand principles and hardware of AR and VR systems.
2. Develop AR/VR applications using Unity XR, ARCore, ARKit, and WebXR.
3. Introduce mathematical foundations of Game Theory and Nash Equilibrium.
4. Apply Algorithmic Game Theory to mechanism design, auctions, and multi-agent systems.

Broad Course Description

A dual-focus course covering immersive XR development (Unity XR Toolkit, ARCore/ARKit, WebXR, spatial audio, hand tracking, Mixed Reality) and Algorithmic Game Theory (Nash Equilibrium, dominant strategies, mechanism design, auction theory, Price of Anarchy, multi-agent coordination). Students build VR environments, AR mobile apps, and game-theoretic simulations.

UNIT – I	06 Hours

Introduction to AR/VR Technologies: History and evolution of AR/VR, VR hardware – HMDs, display technologies, tracking, VR perception – stereoscopic vision, FOV, refresh rates, AR hardware – smartphones, smart glasses, MR and Reality-Virtuality Continuum, Presence, cybersickness, XR design principles.

• Jason Jerald, "The VR Book: Human-Centered Design for VR" Ch. 1–4 (ACM Books, 2016)
• Unity XR Documentation (docs.unity3d.com/Manual/XR.html)

UNIT – II	06 Hours

VR Development with Unity XR: Unity XR Toolkit – XR Rig, controllers, input actions, Room-scale vs. stationary VR, Locomotion – teleportation, smooth locomotion, Interaction systems, Spatial audio, Hand tracking, VR optimization, Deploying to Meta Quest.

• Unity XR Interaction Toolkit Documentation (docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit)
• Meta Quest Developer Hub Documentation (developer.oculus.com)

UNIT – III	06 Hours

AR Development with ARCore/ARKit & WebXR: ARCore/ARKit fundamentals – plane detection, image tracking, face tracking, Unity AR Foundation, Markerless AR, Marker-based AR with Vuforia, AR UI/UX, WebXR with A-Frame/Three.js, Location-based AR, Deployment.

• AR Foundation Documentation (docs.unity3d.com/Packages/com.unity.xr.arfoundation)

• A-Frame Documentation (aframe.io/docs)	
UNIT – IV	**06 Hours**
Foundations of Game Theory: Strategic interactions, Normal/extensive form games, Dominant strategies, Nash Equilibrium – pure and mixed strategies, Zero-sum games and minimax, Cooperative game theory – coalitions, Shapley value, Sequential games, Evolutionary game theory. • Martin Osborne & Ariel Rubinstein, "A Course in Game Theory" Ch. 1–6 (MIT Press, 1994 – free at osborne.economics.utoronto.ca) • Nisan et al., "Algorithmic Game Theory" Ch. 1–3 (Cambridge University Press, 2007)	
UNIT – V	**06 Hours**
Algorithmic Game Theory & Applications: Mechanism design – revelation principle, incentive compatibility, Social choice theory, Arrow's theorem, Auction theory – Vickrey auction, combinatorial auctions, Price of Anarchy, Algorithmic mechanism design, AdWords auctions, P2P networks, Blockchain consensus, Multi-agent AI coordination. • Nisan et al., "Algorithmic Game Theory" Ch. 4–9 • Tim Roughgarden, "Twenty Lectures on Algorithmic Game Theory" Lectures 1–8 (Cambridge, 2016)	

NVIDIA Deep Learning Institute (DLI) Course Mapping

The following NVIDIA DLI courses are recommended to supplement the subject. Free courses are accessible at learn.nvidia.com. Paid instructor-led workshops (ILT) are available via NVIDIA partner institutions.

UNIT	NVIDIA DLI Course Name	Type	Duration	URL / Code
Units 1–2	How to Build a Native OpenUSD XR Application Self-paced: build XR applications using OpenUSD and NVIDIA Omniverse – directly maps to Unit 1–2 XR development principles.	Self-Paced	2 Hrs	FREE DLI+S-OV-17+V1
Unit 2	Build Beautiful, Custom UI for 3D Tools on NVIDIA Omniverse 3D tool UI design –	Self-Paced	2 Hrs	FREE
	directly applicable to VR/XR UI/UX design in Unit 2.			DLI+S-OV-12+V1
Unit 2	Building a 3D Product Configurator With OpenUSD and Omniverse Interactive 3D product visualization – mirrors VR environment interaction and object placement covered in Unit 2.	Self-Paced	2 Hrs	FREE DLI+S-OV-16+V1
Unit 3	Getting Started With Image Segmentation Image segmentation using deep learning – applicable to AR plane detection, object recognition, and occlusion in Unit 3.	Self-Paced	2 Hrs	FREE DLI+C-FX-04+V1
Unit 3	Disaster Risk Monitoring Using Satellite Imagery Real-world computer vision for location-based AR – extends Unit 3 location-based AR with spatial data processing.	Self-Paced	2 Hrs	FREE DLI-SP-DRMS
Units 3–5	Building Real-Time Video AI Applications Building real-time video AI pipelines – relevant to AR camera-based recognition	Self-Paced	2 Hrs	FREE DLI-SP-BRVAA

	and object tracking in Unit 3.			
Units 4–5	Introduction to Graph Neural Networks GNNs for relational data – directly relevant to network game theory, graph-based mechanism design, and multi-agent systems.	Self-Paced	2 Hrs	FREE DLI-SP-IGNN
Units 4–5	Exploring Adversarial Machine Learning Adversarial ML – maps directly to game-theoretic concepts (adversarial players, minimax strategies) in Units 4–5.	Self-Paced	2 Hrs	FREE DLI-SP-EAM

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI in XR Environments: Integrate AI NPCs with natural language in VR using LLM APIs.
- Game-Theoretic AI Analysis: Analyze AlphaGo/AlphaZero strategic reasoning.
- Procedural AR Content: Use AI image generation to create AR target images.
- Reflective Mechanism Design: Design incentive mechanism for AI multi-agent system.
- Ethics of Immersive AI: Surveillance, manipulation, and psychological effects of AI-enhanced XR.

Applications

- Medical Training and Surgical Simulation in VR
- Architectural and Real-Estate VR Walkthroughs
- AR-Based Retail Try-On
- Internet Advertising and Programmatic Auctions
- Blockchain Protocol Design
- Multi-Agent AI Systems

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted

debugging. Lab Experiments

- Set up Unity XR Toolkit; build VR scene with grabbable objects and teleportation.
- Implement hand tracking in VR for natural interaction.
- Build spatial audio-enhanced VR environment.
- Create marker-based AR app overlaying 3D models on image targets.
- Develop markerless AR app for placing virtual furniture in real room.
- Build WebXR experience in browser using A-Frame.
- Implement and solve Nash Equilibrium for 3x3 payoff matrix.
- Simulate Vickrey auction and verify truthful bidding.
- Model network routing game and compute Price of Anarchy.
- Develop multi-agent simulation using game-theoretic coordination strategies.

Mini Project Component

Industry-driven projects that simulate real software engineering

problems. Project 1 – Immersive VR Museum Experience

Problem: VR museum where users walk around, pick up artifacts, and hear spatial audio narration.

Concepts used: Unity XR, Interaction Toolkit, Spatial Audio

Industry relevance: Cultural heritage, tourism, and educational

VR. Project 2 – AR Product Visualization App

Problem: AR mobile app for placing and visualizing 3D products in real environment.

Concepts used: AR Foundation, ARCore/ARKit, 3D Modeling, UI/UX

Industry relevance: E-commerce, retail, and interior design.

Project 3 – Mechanism Design Simulation

Problem: Design and simulate a spectrum auction using algorithmic mechanism design.

Concepts used: Game theory, Auctions, Algorithm design, Agent modeling

Industry relevance: Telecom auctions, digital advertising, policy design.

Course Outcomes

CO	Description	Bloom's
At the end of the course the student will be able to:		
CO1	Explain hardware, perceptual principles, and design guidelines for AR and VR immersive experiences.	L2
CO2	Build interactive VR applications using Unity XR Toolkit with locomotion, interaction, and spatial audio.	L3
CO3	Develop AR applications for mobile platforms using AR Foundation, ARCore/ARKit, and WebXR.	L3
CO4	Analyze strategic interactions using Nash Equilibrium, dominant strategies, and cooperative game theory.	L4
CO5	Apply algorithmic mechanism design principles to design and evaluate auction systems and multi-agent coordination.	L5

Table: Mapping Levels of COs to POs / PSOs												
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	2	2	3						2	2	2
CO2	3	3	3	3	3						2	3	3
CO3	3	3	3	3	3						2	3	3
CO4	3	3	2	3	2						2	3	2
CO5	3	3	3	3	3						3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)													

Suggested Tools

- Unity 2022 LTS with XR Toolkit
- AR Foundation (ARCore/ARKit)
- Vuforia
- A-Frame / Three.js
- Meta Quest / HTC Vive
- Python / Jupyter for Game Theory

TEXT BOOKS:

1. Jason Jerald, "The VR Book", ACM Books, 2016.
2. Nisan et al., "Algorithmic Game Theory", Cambridge University Press, 2007.
3. Martin Osborne & Ariel Rubinstein, "A Course in Game Theory", MIT Press, 1994.

REFERENCE BOOKS:

1. Tim Roughgarden, "Twenty Lectures on Algorithmic Game Theory", Cambridge, 2016.
2. Unity XR Official Documentation (docs.unity3d.com).

E-RESOURCES:

- Unity Learn – XR Development (learn.unity.com)
- Stanford CS364A – Algorithmic Game Theory (YouTube)
- WebXR Samples (immersive-web.github.io)
- NVIDIA DLI: learn.nvidia.com

Activity Based Learning (Suggested Activities in Class)

- VR Demo Day – experience and critique commercial VR applications.
- Game theory debates – simulate multi-player strategic games.
- Interdisciplinary workshops with economics and design students.

****

Quantum Artificial Intelligence

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30hrs L + 15hrs J
L–T–P-J	:	2–0–0-2			

Prerequisites: Linear Algebra, Probability, Machine Learning

Course Learning Objectives:

This Course will enable students to:

- Understand the principles of Quantum Artificial Intelligence and intelligent computational models.
- Analyze information theory, reversible computation, and probabilistic reasoning for intelligent systems.
- Apply quantum-inspired techniques for problem solving and intelligent decision systems.
- Design and simulate Quantum AI models using modern quantum frameworks.
- Explore ethical, explainable, and emerging trends in Quantum Artificial Intelligence.

Course Vision:

To provide students with the ability to integrate quantum computing principles with artificial intelligence for solving complex, large-scale problems. To foster innovation and ethical responsibility in designing quantum-enhanced intelligent systems that address engineering, scientific, and societal challenges.

Broad Course Description

This course introduces the concepts, principles, and applications of Quantum Artificial Intelligence (QAI) by integrating artificial intelligence paradigms with quantum computational models. The course focuses on quantum-enhanced intelligent systems, probabilistic reasoning, quantum cognition, reversible computation, quantum problem solving, and hybrid quantum–classical AI systems. Students will explore emerging Quantum AI approaches through simulation-driven learning and project-based activities using modern quantum platforms.

UNIT – I	08 Hours

Computation and Problem Solving Foundations:

Introduction to Quantum Artificial Intelligence, Artificial Intelligence Foundations, Computational Models, Complexity Theory, Church–Turing Thesis, Classical and Quantum Computation, Knowledge Representation, Production Systems, Deduction Systems, Sub-symbolic Models of Problem Solving.

(Text Book-1: Chapter 1, 2)

UNIT – II	10 Hours

Information, Probability, and Reversible Computation:

Information Theory and Thermodynamics, Entropy and Information Measures, Information and Memory, Associative Memory, Sparse Coding, Reversible Computation, Reversible Circuits, Toffoli Gate, Kolmogorov Probabilities, Conditional Probability, Bayes Rule, Bayesian Networks, Markov Chains. (Text Book-1: Chapter 3, Chapter 4)

UNIT – III	09 Hours

Qubit-Based Computation: Unitary Evolution, Hilbert Space, Quantum Time Evolution, Compound Systems, Von Neumann Entropy, Quantum Measurement, Quantum Randomness, Quantum Circuits, Quantum Fourier Transform, Quantum Parallelism. (Text Book 1: Chapter 5, Chapter 6)	
UNIT – IV	**09 Hours**
Quantum Search and Quantum Problem Solving: Overview of Grover's Search Algorithm, Quantum Search Space Representation, Periodicity and Quantum Fourier Transform, Quantum Problem-Solving Models, Quantum Production Systems, Quantum Search in AI, Quantum Decision Models, Quantum Associative Memory. (Text Book1: Chapter 7, Chapter 8)	
UNIT – V	**09 Hours**
Quantum Cognition and Emerging Quantum AI Approaches: Quantum Cognition, Quantum Decision Theory, Quantum-Inspired Learning Systems, Quantum-Like Bayesian Networks, Hybrid Quantum–Classical AI Models, Explainable and Responsible Quantum AI, Emerging Trends in Quantum Artificial Intelligence. (Text Book1: Chapter 9, Chapter 10)	

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

1. **AI–Quantum Convergence:** Integrate classical AI techniques with quantum computational paradigms to develop advanced intelligent systems.
2. **Quantum Data Representation & Learning:** Utilize quantum-inspired information representation for intelligent processing and reasoning.
3. **Quantum Circuit-Based Intelligence:** Apply quantum computational models for intelligent learning and optimization.
4. **Simulation-Driven AI Development:** Use Qiskit and quantum simulators for experimentation and validation.
5. **Hybrid Quantum–Classical AI Systems:** Combine classical optimization algorithms with quantum computational approaches.
6. **Responsible & Explainable Quantum AI:** Promote transparency, explainability, and ethical practices in Quantum AI systems.

Suggested Tools:

- Tensorflow Quantum
- PennyLane
- Google Colab
- IBM Quantum Expereince
- Qiskit Machine Learning

Applications:

- Intelligent decision support systems

- Quantum-inspired optimization systems
- Healthcare analytics and diagnosis
- Cybersecurity and secure communication
- Financial intelligence and forecasting
- Smart robotics and autonomous systems
- Quantum-enhanced recommendation systems
- Quantum Natural Language Processing

Mini Project Component

Industry-Driven Projects (IBM / Google Quantum AI Style)

These projects simulate real-world **quantum machine learning and intelligent system applications**.

Project 1 – Quantum Decision Intelligence System

6. **Problem:** Develop a quantum-inspired intelligent decision-making systems
7. **Concepts used:** Quantum probability, Bayesian reasoning, optimization
8. **Industry relevance:** Smart analytics, finance, healthcare

Project 2 – Hybrid Quantum–Classical AI Framework

1. **Problem:** Build a hybrid intelligent model integrating classical AI with quantum computation
2. **Concepts used:** Hybrid systems, optimization, Qiskit
3. **Industry relevance:** Predictive analytics, automation

Project 3 – Quantum Search-Based Problem Solver

1. **Problem:** Implement a quantum-inspired search framework for intelligent problem solving.
2. **Concepts used:** Grover’s search, search space representation
3. **Industry relevance**: Robotics, recommendation systems, AI search applications

Course Outcome	Description	Bloom’s Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the foundations of Quantum Artificial Intelligence and intelligent computational models.	L2
2	Analyze information theory, reversible computation, and probabilistic reasoning in intelligent systems.	L4
3	Apply quantum-inspired approaches for intelligent problem solving and decision making.	L3
4	Design and simulate Quantum AI systems using quantum computing frameworks.	L5
5	Analyze emerging applications, ethical aspects, and future directions of Quantum Artificial Intelligence.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	1	1	1							2	
CO2	3	3	2	2	3							3	2
CO3	2	3	2	3	2							3	2
CO4	2	3	2	3	2							3	2
CO5	3	3	3	3	3						2	3	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Andreas Wichert, "**Principles of Quantum Artificial Intelligence**", World Scientific Publishing.

REFERENCE BOOKS:

1. Chris Bernhardt, "Quantum Computing for Everyone", MIT Press.
2. Maria Schuld and Francesco Petruccione, "Supervised Learning with Quantum Computers", Springer.
3. Michael Nielsen and Isaac Chuang, "Quantum Computation and Quantum Information", Cambridge University Press.

E-Resources:

- https://qiskit.org
- https://pennylane.ai
- https://quantum.ibm.com
- https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Quantum Circuit simulation

Activity 2: Analysis of real-world Quantum AI applications in healthcare, finance, cybersecurity, and robotics.

Activity 3: Activities related to entropy, probabilistic reasoning, Bayesian inference, and information measures.

****

Sustainable Innovation Business

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P-J	:	3–0–0-0			

Course Vision:

The course aims to develop knowledge and strategic understanding of innovation management for fostering creativity, competitiveness, and sustainable growth in organizations. It enables students to explore various innovation approaches, emerging business models, and innovation ecosystems while understanding the role of policy, funding, and organizational structures in driving innovation. The course also equips learners with practical frameworks and analytical skills to manage innovation effectively and create value in real-world and emerging market environments.

Prerequisites: basic understanding of management principles, business environment, and organizational behaviour

Course Learning Objectives:

This Course will enable students to:

1. Develop an understanding of fundamental concepts, principles, and processes of innovation.
2. Enable students to analyze the role of innovation in organizational strategy and competitiveness.
3. Familiarize learners with different types and approaches to innovation, including disruptive and open innovation.
4. Provide insights into innovation ecosystems, including policy, funding, and barriers.
5. Equip students with frameworks and tools to manage innovation in real-world contexts, especially emerging markets.

Broad Course Description

This course introduces students to the theory and practice of innovation management in contemporary organizations. It covers the evolution of innovation, its strategic importance, and the processes through which innovation is generated and managed. The course explores different types of innovation, including incremental, disruptive, and design-based innovation, along with modern approaches such as open innovation, reverse innovation, and Jugaad innovation. It also examines the role of government policies, venture capital, and organizational structures in fostering innovation. Through frameworks, models, and real-world examples, the course prepares students to understand, analyze, and apply innovation strategies for sustainable competitive advantage and value creation.

UNIT – I	09 Hours

Foundations of Innovation:

Historical Perspectives of Innovation, Innovation Myths and Realities, Challenges, Triggers, and Sources of Creativity & Innovation, Innovation by Individuals, Communities, and Corporations, Innovator Profiles.

UNIT – II	09 Hours

Innovation Process and Strategy:

Innovation Cycle and Phases, Structured vs. Unstructured Innovation, Link between Corporate Vision, Strategy, and Innovation, Components of Strategic Innovation, Organizational Architecture for Strategic Innovation.

UNIT – III	09 Hours

Innovation Ecosystem:

Role of Government Policy in Innovation, Role of Venture Capitalists and Business Angels, Barriers to

Innovation (Eight Barriers), Principles for Breaking Innovation Barriers, Innovation for Sustainable Competitive Advantage, Wealth and Value Generation	
UNIT – IV	**09 Hours**
Innovation Types and Approaches: Incremental, Breakthrough, and Disruptive Innovation, Design-based Innovation, Open vs Closed Innovation, Kotler's Four Levels of Innovation.	
UNIT – V	**09 Hours**
Innovation Frameworks and Emerging Models: Hansen & Birkinshaw, Treacy & Wiersema, Sawhney & Wolcott, Reverse Innovation: Principles and Practices, Management Transformation, Jugaad Innovation: Principles and Practices, Emerging Market Relevance	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Enabled Innovation Ideation & Validation:**
Leverage AI tools to generate, refine, and validate innovative ideas using market insights, user needs, and emerging trends while ensuring originality and contextual relevance.
- **AI-Assisted Strategic Decision Making:**
Apply AI to analyze data, evaluate innovation opportunities, and support strategic decisions related to product development, market entry, and competitive positioning.
- **Responsible & Ethical Use of AI in Innovation:**
Ensure ethical use of AI by validating outputs, avoiding over-reliance, maintaining transparency, and critically reflecting on biases, limitations, and the role of human judgment in innovation processes.

Tools:
- Miro
- Notion
- Figma

Applications:
- Product and service innovation
- Business model innovation
- Process optimization and automation
- Customer experience enhancement
- Predictive innovation strategies
- Sustainable and frugal innovation

Project Component
- Develop a disruptive business model for a traditional industry using AI.
- Design a low-cost (Jugaad) innovation for rural or underserved markets.
- Develop a smart workflow system using automation tools.
- Analyze successful reverse innovation cases and propose improvements.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain fundamental concepts, principles, and processes of innovation.	L2
2	Analyze the relationship between innovation, corporate strategy, and	L4

	organizational performance.	
3	Evaluate different types and approaches of innovation in various contexts.	L5
4	Assess the role of innovation ecosystems, including policy, funding, and barriers.	L4
5	Apply innovation frameworks and models to solve real-world business problems.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	2	2	3				2	2	2		2
CO2	3	3	2	2	3				2	2	2		2
CO3	2	3	3	2	3				2	2	2		2
CO4	2	2	3	3	3				2	2	2		2
CO5	3	3	2	2	3				2	2	2		2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Paul Trott, *Innovation Management and New Product Development*, Pearson Education, 6th Edition, 2017

REFERENCE BOOKS:

2. Joe Tidd and John Bessant, Managing Innovation: Integrating Technological, Market and Organizational Change, Wiley, 5th Edition, 2018. ISBN – 978-1118360637.

3. Clayton M. Christensen, The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail, Harvard Business Review Press, 2016. ISBN – 978-1633691780.

4. Vijay Govindarajan and Chris Trimble, Reverse Innovation: Create Far From Home, Win Everywhere, Harvard Business Review Press, 2012. ISBN – 978-1422157640.

E-RESOURCES:

- https://www.strategyzer.com
- https://www.ideou.com/blogs/inspiration/what-is-design-thinking

- https://hbr.org/topic/innovation
- https://online.hbs.edu/blog/post/types-of-innovation
- https://www.nesta.org.uk/toolkit/innovation-methods/

- https://www.wipo.int/about-ip/en/

Activity Based Learning (Suggested Activities in Class)

1. Identify real-world problems and propose innovative solutions with field exposure.
2. Develop and prototype an innovative product/service and test with potential users.
3. Analyze real-world case studies of successful and failed innovations.

MOOC Courses

Sl. No.	Course Code	Course Title
1	*CID	Blockchain and its Applications, IIT Kharagpur [NPTEL] 12 WEEKS Blockchain and its Applications | SWAYAM https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs34

Note: CID = NPTEL Course ID

Innovation for Sustainable Business Management

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**:	03
Hours / Week	:	02 Hours	**Total Hours**:	30Hrs L+ 15 Hrs J
L–T–P-J	:	2–0–0-2		

Course Vision:

The course aims to develop an understanding of innovation management principles, strategies, and processes for fostering creativity and competitive advantage in modern organizations. It enables students to analyze different forms of innovation, innovation ecosystems, and emerging business models in dynamic market environments. The course also equips learners with practical frameworks and strategic thinking skills to manage innovation, drive value creation, and support sustainable growth in organizations and emerging economies.

Prerequisites: basic understanding of management principles, business environment, and organizational behaviour

Course Learning Objectives:

This Course will enable students to:

1. Develop an understanding of fundamental concepts, principles, and processes of innovation.
2. Enable students to analyze the role of innovation in organizational strategy and competitiveness.
3. Familiarize learners with different types and approaches to innovation, including disruptive and open innovation.
4. Provide insights into innovation ecosystems, including policy, funding, and barriers.
5. Equip students with frameworks and tools to manage innovation in real-world contexts, especially emerging markets.

Broad Course Description

This course introduces students to the theory and practice of innovation management in contemporary organizations. It covers the evolution of innovation, its strategic importance, and the processes through which innovation is generated and managed. The course explores different types of innovation, including incremental, disruptive, and design-based innovation, along with modern approaches such as open innovation, reverse innovation, and Jugaad innovation. It also examines the role of government policies, venture capital, and organizational structures in fostering innovation. Through frameworks, models, and real-world examples, the course prepares students to understand, analyze, and apply innovation strategies for sustainable competitive advantage and value creation.

UNIT – I	06 Hours

Foundations of Innovation:

Historical Perspectives of Innovation, Innovation Myths and Realities, Challenges, Triggers, and Sources of Creativity & Innovation, Innovation by Individuals, Communities, and Corporations, Innovator Profiles.

UNIT – II	06 Hours

Innovation Process and Strategy:

Innovation Cycle and Phases, Structured vs. Unstructured Innovation, Link between Corporate Vision, Strategy, and Innovation, Components of Strategic Innovation, Organizational Architecture for Strategic Innovation.

UNIT – III	06 Hours

Innovation Ecosystem:

Role of Government Policy in Innovation, Role of Venture Capitalists and Business Angels, Barriers to Innovation (Eight Barriers), Principles for Breaking Innovation Barriers, Innovation for Sustainable Competitive Advantage, Wealth and Value Generation	
UNIT – IV	**06 Hours**
Innovation Types and Approaches: Incremental, Breakthrough, and Disruptive Innovation, Design-based Innovation, Open vs Closed Innovation, Kotler's Four Levels of Innovation.	
UNIT – V	**06 Hours**
Innovation Frameworks and Emerging Models: Hansen & Birkinshaw, Treacy & Wiersema, Sawhney & Wolcott, Reverse Innovation: Principles and Practices, Management Transformation, Jugaad Innovation: Principles and Practices, Emerging Market Relevance	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Enabled Innovation Ideation & Validation:**
  Leverage AI tools to generate, refine, and validate innovative ideas using market insights, user needs, and emerging trends while ensuring originality and contextual relevance.
- **AI-Assisted Strategic Decision Making:**
  Apply AI to analyze data, evaluate innovation opportunities, and support strategic decisions related to product development, market entry, and competitive positioning.
- Responsible & Ethical Use of AI in Innovation:
  Ensure ethical use of AI by validating outputs, avoiding over-reliance, maintaining transparency, and critically reflecting on biases, limitations, and the role of human judgment in innovation processes.

Suggested Tools:

- Miro
- Notion
- Figma

Applications:

- Product and service innovation
- Business model innovation
- Process optimization and automation
- Customer experience enhancement
- Predictive innovation strategies
- Sustainable and frugal innovation

Project Component

- Develop a disruptive business model for a traditional industry using AI.
- Design a low-cost (Jugaad) innovation for rural or underserved markets.
- Develop a smart workflow system using automation tools.
- Analyze successful reverse innovation cases and propose improvements.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Explain fundamental concepts, principles, and processes of innovation.	L2
CO2	Analyze the relationship between innovation, corporate strategy, and organizational performance.	L4
CO3	Evaluate different types and approaches of innovation in various contexts.	L5
CO4	Assess the role of innovation ecosystems, including policy, funding, and barriers.	L4
CO5	Apply innovation frameworks and models to solve real-world business problems.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	2	3			2	2	2	2		3
CO2	3	3	2	2	3			2	2	2	2		3
CO3	2	3	3	2	3			2	2	2	2		3
CO4	2	2	3	3	3			2	2	2	2		3
CO5	3	3	2	2	3			2	2	2	2		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Paul Trott, *Innovation Management and New Product Development*, Pearson Education, 6th Edition, 2017.

REFERENCE BOOKS:

1. Joe Tidd and John Bessant, Managing Innovation: Integrating Technological, Market and Organizational Change, Wiley, 5th Edition, 2018. ISBN – 978-1118360637.
2. Clayton M. Christensen, The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail, Harvard Business Review Press, 2016. ISBN – 978-1633691780.
3. Vijay Govindarajan and Chris Trimble, Reverse Innovation: Create Far From Home, Win Everywhere, Harvard Business Review Press, 2012. ISBN – 978-1422157640.

E-RESOURCES:

- https://www.strategyzer.com
- https://www.ideou.com/blogs/inspiration/what-is-design-thinking
- https://hbr.org/topic/innovation
- https://online.hbs.edu/blog/post/types-of-innovation
- https://www.nesta.org.uk/toolkit/innovation-methods/
- https://www.wipo.int/about-ip/en/

Activity Based Learning (Suggested Activities in Class)

1. Identify real-world problems and propose innovative solutions with field exposure.
2. Develop and prototype an innovative product/service and test with potential users.
3. Analyze real-world case studies of successful and failed innovations.

****

Product Analytics

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code :		**Credits**:	03
Hours / Week :	02 Hours	**Total Hours**:	30Hrs L+ 15 Hrs J
L–T–P–J :	2–0–0–2		

Course Vision:

The course aims to develop a strong foundation in product analytics and data-driven decision-making for understanding user behaviour and improving digital product performance. It enables students to analyze product metrics, user journeys, and customer interactions using modern analytics tools and techniques. The course also equips students with skills in A/B testing, customer segmentation, and personalization to optimize user experience and support strategic product development in modern businesses.

Course Learning Objectives:

This Course will enable students to:

1. Analyse and understand User Behaviour.
2. Explore the skills to leverage product analytics for data driven decision making.
3. Get the idea of tools and technologies around Product Analytics.
4. Describe and illustrate the ways to improve Product performance and optimize user experience.

Broad Course Description

This course introduces Product Analytics and explains how data helps improve product decisions in modern businesses. It covers basic concepts such as metrics, KPIs, and the role of product analysts, along with methods for collecting and analyzing data using tools like Google Analytics, while ensuring data accuracy and privacy.

The course also focuses on understanding user behaviour through techniques like journey mapping, funnel analysis, and cohort analysis. Students learn A/B testing, customer segmentation, and personalization to optimize products. By the end, learners will be able to analyze data, gain insights, and make effective decisions to enhance product performance.

UNIT – I	06 Hours

Introduction to Product Analytics

- Understanding the significance of product analytics in modern business.
- Key concepts: metrics, KPIs, and data-driven decision-making.
- Role of Product Analysts and their responsibilities.

Resources:

Article: “A Beginner’s Guide to Product Analytics” by Mixpanel
Video: “Introduction to Product Analytics” by Heap

UNIT – II	06 Hours

Data Collection and Tools

- Exploring data sources and collection methods.
- Introduction to Google Analytics for tracking user behaviour.
- Ensuring data accuracy, privacy, and compliance.

Resources:

- Google Analytics Academy (Free online courses).
- Article: “Getting Started with Google Analytics” on Analytics Help

UNIT – III	06 Hours

Defining Metrics and KPIs
Defining meaningful metrics and KPIs.
Building a measurement framework for effective analysis. Practical exercise: Identifying KPIs for a social media platform.
Resources:
Article: "Choosing the Right Product Metrics" by Amplitude.

UNIT – IV	06 Hours

User Behaviour Analysis
- Mapping user journeys and funnel analysis.
- Cohort analysis to measure user retention.
- Hands-on: Analyzing user behaviour using cohort analysis.

Resources:
Article: "Cohort Analysis in Google Analytics" by Loves Data
Article: "User Funnel Analysis: A Step-by-Step Guide" by Mixpanel

UNIT – V	06 Hours

A/B Testing and Experimentation, Customer Segmentation, Personalization, Performance
- Understanding A/B testing principles.
- Practical guide to A/B testing with Google Optimize.
- Hands-on: Designing and running an A/B test.

Resources:
Google Optimize (Free tool for A/B testing)
Article: "A/B Testing: A Comprehensive Guide" by CXL
- Segmentation strategies for targeted analysis.
- Leveraging data for personalized user experiences.
- Case studies of successful personalization.

Resources:
Article: "The Power of Personalization in Product Development" by SmarterHQ.
Case Study: "How Netflix Uses Segmentation for Personalization" by Kissmetrics.
- Measuring product success and health.
- Churn analysis and strategies for reducing churn.
- Sentiment analysis of user feedback.

Resources:
Tool: Amplitude for product health measurement.
Article: "Mastering Customer Churn Analysis" by Mixpanel.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)
AI-Assisted Data Analysis:
Use AI tools to analyze product data, identify trends, and generate insights for decision-making.

- **AI in User Behaviour Understanding:**
  Apply AI to study user journeys, segmentation, and personalization while validating results with human judgment.

- **AI-Driven Experimentation:**
  Utilize AI for designing and analyzing A/B tests and optimizing product performance.
- **Critical Evaluation of AI Outputs:**
  Encourage students to assess the accuracy, bias, and reliability of AI-generated insights.

- **Ethical and Responsible AI Use:**
  Promote awareness of data privacy, security, and ethical considerations in analytics.
- **Reflective Learning Practices:**
  Maintain reflection logs to evaluate how AI supports learning and enhances independent analytical thinking.

Suggested Tools:

- Microsoft Copilot
- Power BI
- Optimizely
- Hotjar
- Notion AI

Applications:

- **User Behaviour Analysis:**
  Analyze user journeys, funnels, and retention patterns to understand how users interact with a product.
- **KPI & Metrics Optimization:**
  Use AI to define, track, and improve key performance indicators for product success.
- **Customer Segmentation & Personalization:**
  Group users based on behaviour and deliver personalized experiences to improve engagement.
  - **A/B Testing & Experimentation:**
    Design and evaluate experiments to compare features and optimize product performance.
  - **Predictive Analytics:**

Forecast user behaviour, churn, and demand using AI-driven models.

Project Component

- User Behaviour Analysis of an E-Commerce Platform Using Product Analytics.
- Design and Evaluation of A/B Testing for Website Feature Optimization.
- Customer Segmentation and Personalization Using Analytics Techniques.
- Cohort Analysis for User Retention in a Mobile Application.
- KPI Framework Development for a Social Media Platform.
- Funnel Analysis and Conversion Optimization for an Online Service.
- Predictive Analysis of Customer Churn Using Product Data.
- Performance Analytics Dashboard for Business Decision Making.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Analyze the role of product analytics, metrics, KPIs, and data-driven decision-making in modern businesses.	L4

CO2	Apply data collection methods and analytics tools to gather, manage, and ensure accurate data.	L3
CO3	Evaluate user behavior using metrics, KPIs, funnel analysis, and cohort analysis to assess product performance.	L5
CO4	Implement A/B testing and experimentation techniques to optimize product features and user experience.	L3
CO5	Create customer segmentation, personalization strategies, and performance insights for business decision-making.	L6

Table: Mapping Levels of COs to POs / PSOs

CO S	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O 1	PS O 2
CO 1	2	2	2	2	3				2	2	2	2	3
CO 2	2	2	2	2	3				2	2	2	2	3
CO 3	2	2	2	2	3				2	2	2	2	3
CO 4	2	2	2	3	3				2	2	2	2	3
CO 5	2	2	2	2	3				2	2	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Alistair Croll & Benjamin Yoskovitz ,Lean Analytics: Use Data to Build a Better Startup Faster (Lean Series) ,1st Edition , O'Reilly Media publisher , February 2024 .

REFERENCE BOOKS:

1. Joanne Rodrigues, Product Analytics: Applied Data Science Techniques for Actionable Consumer Insights, 1st Edition, Addison-Wesley publisher, December 2020.
2. Florian Bonnet, The Power of Analytics: Understand your customers and build product and experiences that they love, Kindle Edition, June 2025.

E-RESOURCES:

1. https://www.coursera.org/learn/data-pipelines-and-sql-for-product-analytics?
2. https://www.udemy.com/course/product-analytics-101/?
3. https://www.elevify.com/en/courses/arts-design-and-communication/product-and-product-design/product-analytics-course-ecf45

Project/ Course Deliverables

1. Presentation
    a. The PoC running
    b. Parameters tracked with alternatives
2. PoC / Prototype, Target Specifications
3. Using Any tool with few parametersBusiness Case for Product Analytics

Activity Based Learning (Suggested Activities in Class)

4. Real world problem solving as a team building soft skills as well as getting exposure to product analytics skills as appropriate.
5. Demonstration of PoC

****

Capstone Project – Phase I

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VII

Subject Code	:		**Credits**	:	4
Hours / Week	:	03 Hours	**Total Hours**	:	60 Hours J
L–T–P–J	:	0–0–0–8			

Course Vision:

The course aims to develop students' ability to apply engineering knowledge and research methodologies to solve real-world problems through collaborative project work. It enhances technical, analytical, problem-solving, teamwork, and innovation skills while encouraging creativity in designing practical engineering solutions. The course also prepares students for professional and research-oriented environments through literature survey, problem identification, project planning, technical documentation, and continuous project development under faculty guidance.

Course Learning Objectives:

This Course will enable students to:

2. To apply theoretical knowledge to solve practical problems.
3. Enhance technical, analytical, and problem-solving skills.
4. Foster teamwork and collaboration skills.
5. Develop creativity and innovation in addressing engineering challenges.
6. To analyze and design the solution to the selected problem statement.

Broad Course Description:

1. Each B. Tech Project must be carried out by a group of students at the Institute. To ensure uniform participation of each student, the group size should be preferably at least 3 but not more than 4 students.
2. Each project activity must be supervised by the faculty members of the Institute. These faculty members are termed Project Guides.
3. In case the project is of multi-disciplinary nature, the Project group can be formed consisting of students from other departments. But there must be at least one student and a project Guide from the department who is offering the Project.
4. The topic proposed by both the guide and the student team should be approved by the department chairman and the department project coordinator to proceed further. A degree of industrial input and involvement will be encouraged and can be facilitated through existing academic-industrial collaborations or by addressing specific topics that are of interest to industrial partners.
5. All projects will be closely supervised by the Project Guide with ongoing feedback and guidance at all stages of the project from conception to completion.

The following criteria will be checked by the department chairman to approve for the project proposal:

a. Department staff as Project guide
   - Ability to provide direction to the student in the chosen field of interest to formulate a suitable title of the project.
   - Ability to design an appropriate strategy and methodology to carry out the Project by the team.
   - Ability to provide and evaluate the strong literature review document for the chosen topic.
   - Ability to train students on paper / technical writing skills.

b. Student Team
   - To be dedicated and committed to work on the project by sharpening the existing and learning new technical skills.
   - To be committed to completing the project and participate in hackathons and project exhibitions.

Phase -1 comprises of Literature Survey, Problem identification, Objectives and Methodology.
There will be CIA evaluation (Project reviews) done by a committee of senior faculty of the department based on the rubrics.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Demonstrate the ability to apply engineering principles to solve real-world problems.	L3
2	Develop innovative thinking and thereby preparing students for Capstone project.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	3	3	3	3	3	3	3	3	3	3	3	3
CO2	3	3	3	3	3	3	3	3	3	3	3	3	3

****

Leadership, Policy & Authentic Intelligence

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	02
Hours / Week	:	02	**Total Hours**	:	30 Hrs L
L–T–P–J	:	2 – 0 – 0 – 0			

Course Vision:

The course aims to develop responsible engineering leaders with a strong understanding of AI ethics, policy, governance, and authentic intelligence for designing trustworthy and socially responsible AI systems. It equips students with leadership, critical thinking, and ethical decision-making skills to address challenges such as bias, fairness, transparency, reliability, and accountability in modern AI technologies. The course also promotes awareness of regulatory frameworks and responsible innovation practices required for sustainable and human-centered intelligent systems.

Prerequisites: Students enrolling in LPAI must satisfy the following prerequisites. No prior policy or ethics coursework is assumed, but technical competence and comfort with ambiguity are essential.

Course Learning Objectives:

Students will:

- Understand the fundamentals of engineering and society leadership
- Apply tools of AI ethics, policy
- Develop governance frameworks
- Analyze ethical, societal, and reliability issues in AI systems

Broad Course Description

This course introduces students to the rapidly evolving field of AI and ethics. It covers foundational concepts such as Policy, regulatory, and governance. Students will gain hands-on experience in analysing, comparing various other regulatory on AI applications. The course integrates NVIDIA DLI resources to expose students to Trustworthy AI workflows in deep learning series. Ethical concerns such as bias, hallucination, and responsible AI deployment are also emphasized.

UNIT – I	10 Hours

INTRODUCTION:

Foundations + Leadership 10 hours | Weeks 1–4

Topics covered

- What is **Authentic Intelligence::**? Beyond artificial intelligence
- Engineering in society: roles, responsibilities, and power
- Systems thinking for engineers — feedback loops and stakeholder mapping
- History of technology governance and the leadership–policy–AI nexus
- Leadership models: transformational, servant, and distributed
- Decision-making under uncertainty and high-stakes environments
- Managing diverse, interdisciplinary teams
- Communicating technical ideas to non-technical stakeholders
- Ethical leadership: when to dissent, when to escalate

Teaching methods

- • Socratic seminar and concept mapping (Week 1)
- • Guest talk: industry or government practitioner (Week 2)
- • Leadership style self-assessment and small group case discussion (Week 3)

Project deliverable

- Deliverable Project brief (individual) — 1-page problem framing with stakeholder map
- Format Individual · 1 page

- Due End of Week 4
- Weight 15% of final grade

UNIT – II	10 Hours

Policy, Governance & AI Ethics 10 hours | Weeks 5–7
Topics covered

- How policy is made: legislative, regulatory, and standards processes
- Key frameworks: EU AI Act, India's DPDP Act 2023, IEEE standards
- Engineers as policy actors: testimony, consultation, design advocacy
- Comparative case: Aadhaar — governance lessons for engineers
- Public interest technology and responsible innovation
- Ethical theories applied to AI: consequentialism, deontology, virtue ethics
- Bias, fairness, and representation in algorithmic systems
- **Authentic Intelligence:** human judgment augmented by AI
- Accountability gaps: who is responsible when AI fails?
- Designing for human dignity, autonomy, and informed consen

Teaching methods

- Policy memo writing workshop and India AI regulatory landscape mapping (Week 5)
- Mock regulatory hearing and comparative analysis paired activity (Week 6)
- Ethics case carousel (3 cases, 3 groups) and bias detection exercise (Week 7)
- AI ethics canvas workshop and Socratic philosophical dialogue (Week 7)

Project deliverable

- Deliverable Policy & ethics brief (team) — 2-page governance recommendation + 1-page
- ethics audit card
- Format Team · 3 pages total + verbal debrief
- Due End of Week 7
- Weight 20% of final grade

UNIT – III	10 Hours

Capstone: Integrated Design Challenge 10 hours | Weeks 8–10
Topics covered

- Revisiting your Unit 1 problem brief — what has changed?
- Integrating leadership, policy, and ethics into one design response
- Stakeholder consultation and co-design methods
- Prototyping governance structures and technical safeguards
- Writing for public audiences: op-eds and policy explainers
- Structuring a policy pitch for decision-makers
- Visual communication: slides, infographics, and data stories
- Structured peer review and iteration under time pressure

Teaching methods

- Problem reframing workshop and capstone kick-off Q&A (Week 8)
- Co-design simulation and structured peer consultation (Week 9)
- Pitch rehearsal and writing workshop (Week 9)
- Structured peer review with instructor micro-consultations (Week 10)
- Public showcase with external jury panel: industry, policy, and academia (Week 10)

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Algorithmic thinking for AI model optimization
- Dynamic programming and greedy strategies in reinforcement learning
- Graph algorithms in neural network and social network connectivity

- Heuristic search for game and AI decision systems
- Responsible algorithmic design (fairness, bias minimization)

Applications

1. Search engines – optimizing query processing and keyword indexing.
2. Routing and navigation systems – computing shortest or fastest paths.
3. Scheduling and resource allocation – optimizing CPU, job, or network tasks.
4. Data compression – efficient encoding using greedy and dynamic algorithms.
5. Artificial intelligence and machine learning – applying search and optimization algorithms.
6. Cybersecurity – cryptographic key generation and attack-path analysis.

Mini Project Component

DLI Resources as Case Studies
These are resources you can analyse to understand how **Authentic Intelligence::** principles play out in practice.
Case Study 1 — Deloitte + DLI (enterprise AI governance at scale) NVIDIA collaborated with Deloitte to shape a DLI curriculum customised to the needs of AI professionals and projects, enabling Deloitte to tap into
the power of their own NVIDIA DGX system to explore novel AI use cases. NVIDIA This is a case study in how a major consulting firm institutionalised AI literacy as a governance function — connecting technical training directly to client-facing AI deployments in regulated industry

Case Study 2 — Booz Allen Hamilton + US Air Force DLI teamed with Booz Allen Hamilton to train employees and government personnel, including members of the U.S. Air Force, providing hands-on training for data scientists to solve challenging problems in healthcare, cybersecurity and defence. NVIDIA Newsroom This illustrates authentic intelligence in high-stakes, mission-critical settings where explainability and accountability are non-negotiable — the military context demands AI that can be
audited and overridden by humans.

Case Study 3 — KENET University Faculty Training (Global South capacity building) The Fundamentals of Deep Learning course provided faculty from different KENET member universities a great opportunity to get up-to-date training. NVIDIA KENET is Kenya's national research and education network. This is a case study in authentic intelligence as inclusion — building AI capacity in African universities so that the Global South participates in defining how AI works, not just consuming AI built elsewhere.

Case Study 4 — Generative AI Teaching Kit with Dartmouth (academic-industry cocreation) The Generative AI Teaching Kit, co-developed with Assistant Professor Sam Raymond from Dartmouth College, explores practical implementation of Generative AI through LLMs, diffusion models, multimodal architectures, distributed training, and accelerating inference. NVIDIA Developer The co-development model itself is a case study — when universities co-author industry training content, it creates a feedback loop where academic critical thinking (including ethics and bias analysis) shapes what engineers learn to build.

Case Study 5 — NeMo Guardrails as governance in code the inclusion of NeMo Guardrails in the Generative AI Teaching Kit curriculum is a case study in authentic intelligence by design. Rather than treating safety as a policy document, NeMo Guardrails allows developers to write topical rails, dialogue flows, and output filters in configuration — making ethical constraints enforceable at the inference layer. Analysing how Guardrails works, what it can and cannot prevent, and who controls the rail definitions is a rigorous case study in the limits and possibilities of technical AI governance.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Apply Authentic Intelligence, systems thinking, and leadership models in socio-technical contexts. (Module 1)	
CO2	Demonstrate communication and teamwork skills for presenting technical ideas to stakeholders. (Module 1)	
CO3	Implement AI policy frameworks and governance principles in real-world scenarios. (Module 2)	
CO4	Apply ethical principles to address bias, fairness, and accountability in AI systems. (Module 2).	
CO5	Develop solutions using leadership, policy, and ethical considerations for real-world challenges. (Module 3)	

Table: Mapping Levels of COs to POs / PSOs												
COS	**Program Outcomes (POs)**										**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	2	3	2	2	1	3	2	2	3	2	2	2	
CO2	-		1	-	-	1	-	2	3	3	-	2	
CO3	2	3	2	2	2	3	2	3	2	2	3	3	
CO4	-	2	-	-	-	3	2	3	1	1	-	2	
CO5	2	3	3	2	2	3	2	3	3	2	3	3	
CO6	-	-	-	-	-	-	-	-	-	-	-	-	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

RECOMMENDED TOOLS

- Programming: Python
- Visualization: Transformers, Lang Chain
- Platforms: NVIDIA DLI, Google Colab
- AI Tools: CUDA, RAPIDS

Additional Resources

1. NVIDIA Learning Resources
o NVIDIA Deep Learning Institute (DLI)
o Generative AI Learning Path
2. Recommended:
o Getting Started with Deep Learning
o Accelerating Data Science Workflows
o CUDA Python Fundamentals

****

Agentic AI and RAG Engineering

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hrs L
L–T–P–J	:	3–0–0-0			

Course Vision:

The course aims to develop strong foundations in Agentic AI, Generative AI, and Retrieval-Augmented Generation (RAG) for building autonomous, intelligent, and knowledge-driven systems. It enables students to understand large language models, transformer architectures, prompt engineering, vector databases, and advanced retrieval techniques for creating reliable AI applications. The course also equips students with the skills to design autonomous AI agents capable of reasoning, planning, decision-making, and task execution in complex real-world environments, supporting applications in intelligent assistants, automation, enterprise AI, and next-generation AI systems.

Prerequisites: Machine Learning, Natural Language Processing.

Course Learning Objectives:

This Course will enable students to:

1. Understand the foundations of Generative AI and LLMs including transformer architecture, prompt engineering techniques, and evaluation metrics for building intelligent systems.
2. Develop Retrieval-Augmented Generation (RAG) pipelines by integrating embeddings, vector databases, and retrieval strategies for knowledge-grounded AI applications.
3. Analyze and implement advanced RAG techniques such as query transformation, re-ranking, self-RAG, and evaluation frameworks to improve accuracy and reliability.

Broad Course Description

Agentic AI refers to intelligent systems capable of acting autonomously to achieve defined goals by perceiving context, reasoning, planning, making decisions, and executing actions with minimal human intervention. Unlike traditional AI models that primarily respond to prompts, Agentic AI systems can operate as goal-driven agents that use tools, APIs, memory, and feedback loops to solve multi-step problems. These systems integrate capabilities such as task decomposition, autonomous decision-making, workflow orchestration, and adaptive learning, enabling applications in automation, software development, intelligent assistants, robotics, cybersecurity, and enterprise operations. Agentic AI represents a shift from reactive AI toward proactive, self-directed, and collaborative intelligent systems.
Retrieval-Augmented Generation (RAG) is an advanced AI framework that enhances large language models by combining information retrieval with generative capabilities. Instead of relying solely on pre-trained knowledge, RAG retrieves relevant information from external data sources such as databases, knowledge graphs, vector stores, or enterprise documents and uses that context to generate more accurate, up-to-date, and domain-specific responses. This approach improves factual grounding, reduces hallucinations, and enables customized knowledge-driven applications such as intelligent search, question answering, chatbots, and decision support systems. RAG typically integrates embeddings, vector databases, retrievers, and generative models to create scalable and trustworthy AI solutions.
Together, Agentic AI and RAG form a powerful foundation for next-generation intelligent systems, where autonomous agents can reason, plan, retrieve relevant knowledge, and execute tasks dynamically in complex environments. Their integration supports advanced applications such as autonomous research assistants, AI copilots, intelligent automation, multi-agent systems, and enterprise knowledge agents, making them central to the future of generative and autonomous AI

UNIT – I	09 Hours

Foundations of Gen AI & LLM:

Introduction to Generative AI: Scope, Applications, and Industry Use Cases, Transformer Architecture: Encoder-Decoder, Self-Attention, Multi-Head Attention, Tokenization Strategies: BPE, WordPiece,

SentencePiece, Pre-training vs Fine-tuning: Concepts and Differences, Prompt Engineering

Fundamentals: Zero-shot, Few-shot, Chain-of-Thought, Overview of Popular LLMs: GPT-4, Claude, LLaMA, Mistral, Gemini, LLM Evaluation Metrics: BLEU, ROUGE, Perplexity, Human Evaluation, Limitations of Base LLMs: Hallucinations, Stale Knowledge, Context Window Constraints, Introduction to LangChain and LlamaIndex Frameworks, Hands-on: Setting up LLM API (OpenAI / Anthropic) and basic prompt interaction.
Textbook 1: Chapter: 1, 2, 3

UNIT – II	09 Hours

Retrieval-Augmented Generation (RAG):
Motivation for RAG: Solving Hallucination and Knowledge Staleness, RAG Pipeline Architecture: Ingestion → Retrieval → Augmentation → Generation, Document Loaders: PDF, HTML, CSV, DOCX using LangChain, Text Chunking Strategies: Fixed-size, Recursive, Semantic Chunking, Embedding Models: OpenAI Embeddings, Sentence-BERT, INSTRUCTOR, BGE, Vector Databases: FAISS, Chroma, Pinecone, Weaviate – Setup and Comparison, Similarity Search: Cosine Similarity, Euclidean Distance, HNSW Indexing, Retrieval Techniques: Dense Retrieval, Sparse (BM25), Hybrid Retrieval, Context Window Management: Stuffing, Map-Reduce, Refine Strategies, Hands-on: Build an end-to-end RAG pipeline over a PDF knowledge base.
Textbook 2: Chapter: 4, 5, 6, 7

UNIT – III	09 Hours

Foundations of Retrieval- Augmented Generation:
Naive RAG vs Advanced RAG vs Modular RAG: Architectural Comparison, Query Transformation: HyDE, Step-back Prompting, Multi-query Retrieval, Re-ranking Strategies: Cross-Encoder Re-ranking, Cohere Rerank, ColBERT, Self-RAG: Adaptive Retrieval with Reflection Tokens, Corrective RAG (CRAG): Dynamic Retrieval Correction, Multi-vector Retrieval: Parent-Child Chunks, Summary Indexing, Contextual Compression and LLM-based Filtering, RAG Evaluation Frameworks: RAGAS, TruLens – Faithfulness, Relevance, Groundedness, Guardrails and Output Validation in RAG Systems, Hands-on: Implement Self-RAG and evaluate using RAGAS on a custom dataset.
Textbook 4: Chapter: 5, 6, 7

UNIT – IV	09 Hours

Agentic AI – Autonomous Agents & Multi Agent System:
Introduction to AI Agents: Definition, Properties, and Cognitive Architecture, ReAct Framework: Reasoning + Acting – Thought-Action-Observation Loop, Tool Use and Function Calling: APIs, Calculators, Web Search, Code Execution, Memory in Agents: Short-term (Conversation Buffer), Long-term (Vector Memory), Episodic Memory, Planning Strategies: Tree-of-Thoughts, Plan-and-Solve, Hierarchical Planning, LangChain Agents: AgentExecutor, Custom Tools, OpenAI Functions Agent, Introduction to Multi-Agent Systems: CrewAI, AutoGen, LangGraph, Agent Roles and Collaboration: Planner, Executor, Critic, Synthesizer, Human-in-the-Loop (HITL): Checkpoints, Interrupts, and Approval Gates, Hands-on: Build a multi-tool Research Agent using LangGraph with web search and RAG.
Textbook 4: Chapter: 1, 2, 3, 4, 5

UNIT – V	09 Hours

Production Deployment, Security & Real-World Applications:
Agentic RAG: Combining Agents with Dynamic Retrieval for Complex Tasks, Fine-tuning vs RAG vs Agents: When to Use Which Strategy, Productionizing RAG: Caching, Async Processing, Load Balancing, LLMOps: Monitoring, Observability with LangSmith, Phoenix Arize, Security in LLM Systems: Prompt Injection, Jailbreaking, Data Privacy, Responsible AI: Bias Detection, Toxicity

Filtering, Output Guardrails (NeMo Guardrails), Cost Optimization: Token Management, Model Routing, Caching Strategies, Real-world Case Studies: Customer Support Bots, Code Assistants, Legal Research, Medical QA, Emerging Trends: Multimodal RAG, Agentic Workflows with Vision Models, Capstone Project: Design and present a full Agentic RAG system for a domain of choice.
Textbook 1: Chapter: 8, 9, 10

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **AI-Assisted Prompt Engineering and Agent Design:** Students use generative AI tools to design prompts, develop agent workflows, and analyze how autonomous agents reason, plan, and execute tasks in real-world scenarios.
- **Retrieval-Augmented Learning with Knowledge Systems:** Students explore RAG frameworks by integrating vector databases, retrievers, and LLMs to understand contextual retrieval, factual grounding, and knowledge-driven response generation.
- **Human-AI Collaboration for Problem Solving:** Learners critically evaluate how humans collaborate with AI agents in decision-making, comparing automated agent outputs with human reasoning to assess accuracy, trust, and reliability..
- **Authentic Reflection on AI Agent Behavior:** Students reflect on agent autonomy, bias, hallucination, and ethical implications while analyzing the strengths and limitations of Agentic AI and RAG systems.
- **Ethical and Responsible AI Reflection:** Students examine privacy, security, governance, and responsible deployment challenges associated with autonomous agents and retrieval-augmented generative systems..

Applications:

1. Autonomous Research Assistants
2. Enterprise Knowledge Assistants
3. AI Copilots for Software Development
4. Healthcare Decision Support Systems
5. Multi-Agent Workflow Automation

Laboratory Component

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging.

Lab Experiments

1. Design a simple autonomous agent to perform goal-based task execution using prompt engineering.
2. Create a retrieval-augmented generation system using embeddings, vector database, retriever, and LLM.
3. Implement document chunking, embedding generation, and similarity-based retrieval.
4. Build a system that retrieves knowledge from custom documents and generates

contextual answers.

5. Design an agent capable of calling APIs or external tools to complete multi-step tasks.

Mini Project Component

- **Industry-Driven Projects**

Some sample projects described below simulate the **real software engineering problems**. **Project 1 – Intelligent Research Assistant using RAG**

- **Problem:** Develop a system that retrieves information from research papers and generates contextual answers for academic queries..
- **Concepts used:** Vector embeddings, document indexing, semantic search, retrieval pipelines, LLM prompting, RAG architecture.
- **Industry relevance:** Used in research automation, knowledge management, digital libraries, and academic assistant systems.

Project 2 – Agentic Customer Support Chatbot

- **Problem:** Build an autonomous AI agent that retrieves company policy data and resolves customer queries through multi-step reasoning
- **Concepts used:** AI agents, tool calling, RAG pipelines, memory, prompt engineering, workflow orchestration.
- **Industry relevance:** Applied in customer support automation, enterprise helpdesks, and intelligent service assistants.

Project 3 – Resume Screening Assistant using Agentic RAG

- **Problem:** Create an AI agent that retrieves job descriptions, evaluates resumes, and ranks candidates automatically.
- **Concepts used:** RAG, autonomous agents, embeddings, semantic matching, reasoning workflows, vector databases.
- **Industry relevance:** Used in HR automation, talent analytics, and intelligent recruitment platforms.

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
CO1	Understand the architecture and working principles of Large Language Models, transformer attention mechanisms, and their role in generative AI systems.	L4
CO2	Design and implement a complete RAG pipeline including document ingestion, embedding generation, vector storage, and context-aware response generation.	L5
CO3	Apply advanced RAG strategies such as re-ranking, query transformation, multi-vector retrieval, and evaluation frameworks to build production-grade RAG systems.	L5
CO4	Design, implement, and orchestrate LLM-powered autonomous agents with tool use, memory, planning, and multi-agent coordination capabilities.	L5

CO5	Deploy, monitor, secure, and optimize Agentic AI and RAG systems in real-world production environments while addressing ethical considerations and compliance requirements.	L5

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2	2	3	2						2	3	2
CO2	3	3	2	3	3						2	3	2
CO3	3	3	2	3	3						2	3	2
CO4	3	3	2	3	3						2	3	2
CO5	2	3	2	3	3						2	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

Suggested Tools:

- Frameworks & Libraries: LangChain, LlamaIndex, LangGraph, CrewAI, AutoGen, HuggingFace Transformers, RAGAS, TruLens, NeMo Guardrails
- LLM & Embedding APIs: OpenAI GPT-4 / GPT-4o, Anthropic Claude API, Cohere Embed & Rerank, Sentence-Transformers, BGE
- Vector DBs & Observability: FAISS, Chroma, Pinecone, Weaviate, Qdrant, LangSmith, Phoenix Arize, Google Colab / Jupyter

TEXT BOOKS:

1. Building LLM Powered Applications – Valentina Alto, Packt Publishing, 2024
2. Generative AI with LangChain (Second Edition): Build Production-Ready LLM Applications and Advanced Agents using Python, LangChain, and LangGraph – Ben Auffarth and Leonid Kuligin, Packt Publishing, 2025.
3. *Developing Apps with GPT-4 and ChatGPT* – Olivier Caelen & Marie-Alice Blete, O'Reilly Media, 2nd Edition, 2024.
4. *AI Agents in Action* – Micheal Lanham, Manning Publications, 2025. ISBN: 978-1633436343.

E-RESOURCES:

1. https://www.deeplearning.ai/short-courses/building-agentic-rag-with-llamaindex/
2. https://python.langchain.com/docs/
3. https://docs.llamaindex.ai/
4. https://developer.nvidia.com/blog/

Activity Based Learning (Suggested Activities in Class)

7. Student seminars (on topics of the syllabus and related aspects (individual activity).
8. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams).
9. Study projects (by very small groups of students on selected local real-time problems about the syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity).

****

Business Intelligence & Analytics

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30Hrs L + 15 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision: The course aims to develop knowledge and skills in business intelligence, data analytics, and decision-support systems for transforming data into meaningful business insights. It enables students to apply data warehousing, machine learning, artificial intelligence, and secure data processing techniques for predictive analytics and strategic decision-making. The course also emphasizes ethical, sustainable, and explainable AI practices in developing intelligent business dashboards and modern enterprise analytics solutions.

Course Learning Objectives:

This Course will enable students to:

1. Understand BI life cycle and data architecture.
2. Apply data preparation and warehousing concepts.
3. Employ ML and AI for predictive analytics and decision support.
4. Implement secure analytics using blockchain verified data.
5. Develop sustainable and ethical BI dashboards.

Broad Course Description

This course focuses on transforming large data sets into actionable business insights through data warehousing, analytics, and visualization techniques. Students apply machine learning, secure data processing, and explainable AI to build intelligent, ethical, and sustainable decision-support systems for modern enterprises.

UNIT – I	06 Hours

BI FOUNDATIONS:

Data driven decision making concepts. BI architecture – data sources, ETL, data marts. Dimensional modeling – Star vs Snowflake schema. OLAP and ROLAP systems.
(Text Book 1 : Ch. 1 1.1–1.4; Text Book 2 : Ch. 2 2.1–2.3)

UNIT – II	06 Hours

DATA PREPARATION AND INTEGRATION:

Data cleaning and transformation techniques. Handling missing values, outlier detection. Integration of heterogeneous data sources. Use of ETL tools (Talend, Informatica, Power Query)
(Text Book 1 : Ch. 3 3.1–3.4)

UNIT – III	06 Hours

ANALYTICS MODELS AND VISUALIZATION:

Descriptive, Predictive, Prescriptive analytics. ML methods – regression, classification, clustering. Visualization tools – Tableau, Power BI, Matplotlib. KPI design and dashboard principles.
(Text Book 1 : Ch. 5 5.1–5.5; Text Book 2 : Ch. 3 3.1–3.4)

UNIT – IV	06 Hours

ADVANCED BI & SECURE ANALYTICS:

Big data platforms (Hadoop, Spark). Real-time analytics and stream processing. Blockchain verified data and data lineage. Explainable AI (XAI) in analytics.
(Text Book 2 : Ch. 4 4.1–4.3)

UNIT – V	06 Hours

ETHICAL AND SUSTAINABLE ANALYTICS: Data ethics and bias awareness. Regulatory compliance (GDPR, HIPAA). Green data centers and energy metrics. AI governance frameworks for analytics solutions. (Text Book 1 : Ch. 7 7.1–7.4)

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Fair and transparent AI models.
- Dashboard interpretability and bias control.

Applications:

○ Designing dashboards for executive decision-making and performance tracking.

○ Applying predictive analytics for market and risk forecasting.

○ Integrating blockchain-verified data into BI pipelines for data integrity.

○ Using machine learning for customer segmentation and trend analysis.

○ Building ethical, explainable AI-driven business decision systems.

Mini Project Component

■ Distributed analytics for multi-hospital data (BI + Blockchain).

■ Energy-aware data center monitoring dashboard.

■ Project: ETL pipeline building for multi sensor datasets.

■ Journal: Draw a BI architecture for hospital IoT use case

■ Journal: Design dashboard for IoT health blockchain data

■ Project: Integrate BI dashboard with smart contract ledger.

■ Capstone: “Responsible AI Business Dashboard” project for smart security ecosystem.

Course Outcome	Description	Bloom’s Taxonomy Level
At the end of the course the student will be able to:		
CO1	Describe architecture and components of BI systems.	L2
CO2	Perform ETL operations and data integration.	L3
CO3	Apply analytics and ML models for decision making.	L3
CO4	Implement secure and auditable analytics using blockchain.	L4
CO5	Design ethical and sustainable BI applications.	L4

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	2			2						2	2	
CO2	2	3	2	2	3						2	3	
CO3	2	3	3	2	3						2	3	
CO4	2	2	3	2	3			2			2	3	
CO5	2	2	3	2	2		2	2			2	3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Ramesh Sharda et al., Business Intelligence and Analytics, Pearson.
2. Thomas H. Davenport, Analytics at Work, Harvard Press.

NVIDIA Course Mapping (Industry Alignment)

NVIDIA DLI Course / Content	Mapped Units	Relevance
Accelerated Data Science with RAPIDS	II–IV	GPU ETL and analytics
Responsible AI Development	V	Ethical AI dashboards

E-RESOURCES:

1. nptel.ac.in
2. coursera.org
3. edx.org
4. learn.microsoft.com
5. developer.nvidia.com

Activity-Based Learning (Suggested Activities in Class)

- Design a simple ETL pipeline for sample CSV data using Power Query.
- Create a Tableau dashboard with multiple KPIs and charts.
- Run a prediction model on sales data using Python or R.
- Present ethical considerations of AI-based business decisions.
- Conduct a group project on real-time data analytics case study.

****

Unreal Engine Game Development for Beginners

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	03
Hours / Week	:	04 Hours	**Total Hours**	:	30 Hrs L + 30 Hrs J
L–T–P-J	:	2–0–0-2			

Course Vision:

The course aims to develop creative and technical skills in designing interactive 3D applications and game environments using Unreal Engine 5. It enables students to build immersive game experiences through Blueprint Visual Scripting, advanced environment design, animation, visual effects, and modern rendering technologies such as Lumen and Nanite. The course also prepares students for careers in game development, virtual production, simulation, AR/VR, and digital content creation using industry-standard tools and workflows.

Course Learning Objectives:

This Course will enable students to:

1. Gain familiarity with Unreal Engine 5 editor, workflow, and core systems.
2. Develop game logic using Blueprint Visual Scripting without C++.
3. Create visually stunning environments using Lumen, Nanite, and Quixel Megascans.
4. Build and deploy complete interactive 3D game experiences.

Broad Course Description

Unreal Engine 5 (UE5) is used for AAA games, virtual production, architectural visualization, and automotive design. This course focuses on Blueprint Visual Scripting, UE5 editor (Actors, Components, World Outliner), environment design (Landscape, Foliage, Water, Sky Atmosphere), Quixel Megascans, Lumen GI, Nanite geometry, UMG UI, MetaSound, Animation Blueprint, Sequencer cinematics, Niagara particles, and Windows/Linux deployment.

UNIT – I	**06 Hours**

Unreal Engine 5 Foundations: UE5 Editor – Viewport, Content Browser, World Outliner, Details Panel, Actors, Components, Levels, Coordinate systems, Importing assets, Blueprint fundamentals – Event Graph, variables, execution flow, UE5 Project templates.

- Luis Cataldi & Andi McClure, "Unreal Engine 5 Beginner's Guide" Ch. 1–3 (Packt, 2023)
- Epic Games, Official UE5 Documentation (docs.unrealengine.com)

UNIT – II	**06 Hours**

Blueprint Visual Scripting: Blueprint Class types – Actor, Pawn, Character, GameMode, Blueprint nodes – Events, Functions, Macros, Variables, Blueprint communication – Direct reference, Event Dispatchers, Interfaces, Casting, Enhanced Input System – Input Mapping Context, Blueprint debugging.

• Joeri Ouwerkerk, "Blueprints Visual Scripting for UE5" Ch. 1–5 (Packt, 2022)
• Luis Cataldi Ch. 4–6

UNIT – III	**06 Hours**

Environment Design & Rendering: Landscape – sculpting, painting, erosion tools, Foliage Tool, Water Body system, Sky Atmosphere, Volumetric Clouds, Quixel Megascans via Bridge, Material Editor – PBR texturing, Material Instances, Lumen – dynamic GI and reflections, Nanite – virtualised geometry, TSR.

- Luis Cataldi Ch. 7–9
- Quixel Megascans Documentation (quixel.com/megascans/documentation)
- NVIDIA RTX/Lumen Comparison Documentation (docs.unrealengine.com/5.0/RealTimeLumenGlobalIllumination)

UNIT – IV	06 Hours
Gameplay Systems & UI: Collision profiles, Pickup/interaction systems, Inventory basics, Game Framework – GameMode, GameState, PlayerController, UMG – Widget Blueprints, HUD design, Main menu/pause menu, MetaSound, Animation Blueprint – State Machine, Blend Spaces, Aim Offset. ● Joeri Ouwerkerk Ch. 6–9 ● UMG UI Designer Documentation (docs.unrealengine.com/5.0/umg-ui-designer)	

UNIT – V	06 Hours
Cinematics, Polish & Deployment: Sequencer – cinematic cutscenes, camera animation, Timeline tracks, Cinematic cameras – DoF/focal length, Chaos physics – destructible meshes/cloth, Niagara particle system, Post-process volumes, Performance optimization – LODs, occlusion culling, Packaging for Windows/Linux. • Luis Cataldi Ch. 10–12 • NVIDIA RTX in Unreal Engine (developer.nvidia.com/rtx/unreal-engine) • Epic Games, "Niagara Visual Effects System" Documentation	

NVIDIA Deep Learning Institute (DLI) Course Mapping

The following NVIDIA DLI courses are recommended to supplement the subject. Free courses are accessible at learn.nvidia.com. Paid instructor-led workshops (ILT) are available via NVIDIA partner institutions.

UNIT	NVIDIA DLI Course Name	Type	Duration	URL / Code
Unit 3	How to Build a Native OpenUSD XR Application Self-paced: building XR/3D applications with OpenUSD – directly	Self-Paced	2 Hrs	FREE DLI+S-OV-17+V1
	relevant to UE5's OpenUSD integration and 3D pipeline in Unit 3.			
Unit 3	Getting Started With USD for Collaborative 3D Workflows USD workflows for 3D collaboration – applicable to UE5's Datasmith/OpenUSD import pipeline and environment design in Unit 3.	Self-Paced	2 Hrs	FREE DLI+S-OV-20+V1
Unit 3	Essentials of USD in NVIDIA Omniverse Essential USD concepts in Omniverse – foundational for UE5 environment pipeline (Quixel, Megascans, USD assets) in Unit 3.	Self-Paced	2 Hrs	FREE DLI+S-OV-15+V1
Unit 3	Building a 3D Product Configurator With OpenUSD and Omniverse Interactive 3D configurators with Omniverse/OpenUSD – applicable to UE5 architectural visualization workflows.	Self-Paced	2 Hrs	FREE DLI+S-OV-16+V1
Units 3–5	Essentials of Developing Omniverse Kit Applications Omniverse Kit development – relevant to NVIDIA-powered real-time rendering and physics in UE5 environment workflows.	Self-Paced	4 Hrs	FREE DLI+S-OV-11+V1
Unit 3	Easily Develop Advanced 3D Layout Tools on NVIDIA Omniverse Advanced 3D layout tools – directly applicable to UE5 environment layout, Level design, and spatial design in Unit 3.	Self-Paced	2 Hrs	FREE DLI-SP-EDATLO

Unit 5	Bootstrapping Computer Vision Models with Synthetic Data ILT: generate synthetic data using Omniverse – relevant to UE5 Sequencer-based synthetic dataset generation for AI training.	Workshop (ILT)	8 Hrs	PAID DLI-ILT-BCVMSD
Units 4–5	Generative AI Explained GenAI overview – applicable to UE5's MetaHuman Creator, AI NPC behavior, and procedural content generation.	Self-Paced	1 Hr	FREE DLI+S-FX-07+V1
Unit 5	Synthetic Data Generation for Training Computer Vision Models Creating photorealistic training datasets using Omniverse – extends UE5's Sequencer and Nanite pipeline for AI training data generation.	Self-Paced	2 Hrs	FREE DLI-SP-SDGCVM
Units 3–5	Build Beautiful, Custom UI for 3D Tools on NVIDIA Omniverse Professional 3D UI design patterns – applies to UE5 UMG widget design for game HUDs, menus, and interactive visualization.	Self-Paced	2 Hrs	FREE DLI+S-OV-12+V1

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- AI-Assisted Environment Generation: Use UE5's PCG framework and AI tools for environments.
- MetaHuman Creator Exploration: AI-powered digital humans and implications for identity.
- Blueprint Code Review with AI: Convert Blueprint logic to C++ using AI assistants.
- AI NPC Design: Compare behavior trees with LLM-driven NPC dialogue systems.
- Responsible Realism: Ethics of photorealistic rendering – deepfakes and digital doubles.

Applications

- AAA Video Game Development (Fortnite, Hellblade II)
- Film and Virtual Production (The Mandalorian)
- Architectural Visualization
- Automotive Design and Virtual Showrooms
- Training Simulations and Defense Applications
- Live Events and Concert Visualizations

Laboratory Component (1 Credit)

The lab emphasizes hands-on implementation, visualization, and AI-assisted debugging. Lab Experiments

- Navigate UE5 editor; build furnished room scene using Starter Content.
- Create Blueprint Actor that rotates, changes color on overlap.
- Implement third-person character with movement/sprint/jump via Enhanced Input.
- Sculpt and paint 500m×500m landscape with foliage and water body.
- Set up outdoor environment with Lumen GI, Sky Atmosphere, volumetric clouds.

• Build pickup system with Blueprint-driven HUD counter.
• Create main menu Widget Blueprint with New Game, Continue, Quit buttons.
• Design enemy Blueprint with patrol behavior using timers and splines.
• Create Sequencer cinematic with character animation, camera movement, audio.
• Package and export playable game level as standalone Windows executable.

Mini Project Component

Industry-driven projects that simulate real software engineering problems.

Project 1 – First-Person Exploration Game

Problem: Photorealistic FP exploration with environmental storytelling, interactive objects, and atmospheric audio.

Concepts used: UE5 environment design, Blueprint interaction, Lumen/Nanite, MetaSound Industry relevance: Walking simulation genre – popular on Steam and itch.io.

Project 2 – Third-Person Adventure Level

Problem: 3rd-person adventure level with enemies, collectibles, boss encounter, and cutscene. Concepts used: Character Blueprint, AI behavior trees, Sequencer, UMG Industry relevance: Action-adventure games – largest commercial game genre. Project 3 – Architectural Visualization Experience

Problem: Photorealistic building walkthrough with real-time lighting and tour animation.

Concepts used: Quixel Megascans, Lumen, Nanite, Sequencer, Material Instances

Industry relevance: Architecture firms, real estate, interior design.

Course Outcomes

CO	Description	Bloom's
At the end of the course the student will be able to:		
CO1	Navigate UE5 editor proficiently and manage Actors, Components, and Level assets in 3D project workflows.	L2
CO2	Implement complete game logic using Blueprint Visual Scripting including character control and game state management.	L3
CO3	Design photorealistic 3D environments using Landscape, Quixel Megascans, Lumen, and Nanite technologies.	L3
CO4	Build gameplay systems including UI/HUD, animation, audio integration, and game framework setup.	L4
CO5	Create cinematic sequences with Sequencer, apply Niagara particles, optimize performance, and deploy packaged game builds.	L5

Table: Mapping Levels of COs to POs / PSOs													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	2	2	3						2	2	2
CO2	3	3	3	3	3						2	3	3
CO3	3	3	3	3	3						3	3	3
CO4	3	3	3	3	3						3	3	3
CO5	3	3	3	3	3						3	3	3

Suggested Tools

- Unreal Engine 5
- Quixel Bridge – Megascans
- MetaHuman Creator
- Blender
- Visual Studio (C++ companion)
- Audacity / Reaper

TEXT BOOKS:

1. Luis Cataldi & Andi McClure, "Unreal Engine 5 Beginner's Guide", Packt, 2023.
2. Joeri Ouwerkerk, "Blueprints Visual Scripting for Unreal Engine 5", Packt, 2022.
3. Epic Games, "Unreal Engine 5 Official Documentation" (docs.unrealengine.com).

REFERENCE BOOKS:

4. Sjoerd De Jong, "Unreal Engine Lighting and Rendering Guide" (Unreal Engine Community Wiki).
5. Tom Shannon, "Unreal Development Kit Game Design", Packt, 2012.

E-RESOURCES:

- Unreal Online Learning (dev.epicgames.com/community/learning)
- Unreal Sensei YouTube Channel
- William Faucher YouTube – UE5 Environment Art
- NVIDIA DLI: learn.nvidia.com

Activity Based Learning (Suggested Activities in Class)

- Unreal Game Jam – 72-hour game creation challenge.
- Environment art showcase – students display polished UE5 scenes.
- Guest lectures from virtual production and architectural visualization professionals.

****

Quantum Cryptography

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	30 Hrs L + 15 Hrs J
L–T–P–J	:	2–0–0-2			

Course Vision:

The course aims to develop a strong foundation in quantum cryptography and secure communication systems by integrating principles of classical cryptography, quantum mechanics, and information theory. It enables students to understand and analyze quantum cryptographic protocols, quantum key distribution techniques, and security mechanisms against modern cyber threats and quantum attacks. The course also prepares students to design future-ready secure communication solutions for applications in cybersecurity, finance, defense, and next-generation communication networks.

Prerequisites: Cryptography, Quantum Computing Systems

Course Learning Objectives:

This Course will enable students to:

1. Understand classical and quantum cryptography fundamentals
2. Learn information theory and quantum information concepts
3. Analyze quantum key distribution and secret key distillation
4. Explore quantum cryptographic protocols and their implementations
5. Evaluate security aspects of quantum communication systems

Broad Course Description

The course Quantum Cryptography provides a comprehensive understanding of secure communication systems by integrating classical cryptographic techniques with quantum information principles. It covers foundational concepts of classical cryptography, information theory, and quantum mechanics, followed by advanced topics such as Quantum Key Distribution (QKD), secret key distillation, and protocols like BB84 and GG02. The course also explores security analysis against eavesdropping and quantum attacks. Overall, it equips students with the knowledge to design secure, future-ready cryptographic systems for applications in cybersecurity, finance, and communication networks.

UNIT – I	08 Hours

Introduction to Classical Cryptography

Introduction, classical cryptography: Confidentiality and secret-key ciphers, Secret-key authentication, Public-key cryptography

(Text Book-1: Chapter 2)

UNIT – II	10 Hours

Information Theory and Quantum Information Theory

Source coding, Joint and conditional entropies, Channel coding, Renyi entropies, Continuous variables, Perfect secrecy revisited. Fundamental definitions in quantum mechanics, Qubits and qubit pairs, Density matrices and quantum systems, Entropies and coding, Particularity of quantum information, Quantum optics.

(Text Book-1: Chapter 3, Chapter 4)

UNIT – III	09 Hours

Cryptosystems based on Quantum Key Distribution and Secret Key Distillation:

A key distribution scheme, a secret-key encryption scheme, Combining quantum and classical cryptography, Implementation of a QKD-based cryptosystem. Secret Key Distillation: A two-step approach, Characteristics of distillation techniques, authenticated one-shot secret-key distillation, Authenticated repetitive secret-key distillation, unauthenticated secret-key distillation, secret-key

distillation with continuous variables. (Text Book 1: Chapter 5, Chapter 6)	
UNIT – IV	**09 Hours**
The BB84 Protocol, and GG02 Protocol Description, Implementation of BB84, Eavesdropping and secret key rate, from discrete to continuous variables, a protocol with squeezed states, a protocol with coherent states: the GG02 protocol, Implementation of GG02, GG02 and secret-key distillation. (Text Book1: Chapter 10, Chapter 11)	
UNIT – V	**09 Hours**
Security Analysis of Quantum Key Distribution Eavesdropping strategies and secret-key distillation, Distillation derived from entanglement purification, Application to the GG02 protocol. (Text Book1: Chapter 12)	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- **Secure Intelligent Systems Design:** Integrate classical and quantum cryptographic techniques to build secure AI systems that ensure confidentiality, integrity, and authentication of data.
- **Information-Theoretic AI Security:** Apply entropy, coding theory, and quantum information principles to enhance data privacy, secure communication, and robust AI model training.
- **Quantum-Safe Cryptographic Models:** Utilize Quantum Key Distribution (QKD) and post-quantum cryptography to protect AI systems from future quantum attacks.
- **AI-Driven Threat Detection:** Leverage AI techniques to identify eavesdropping strategies, anomalies, and potential attacks in quantum communication systems.
- **Reliable and Noise-Aware Security Systems:** Incorporate quantum noise models and error handling in cryptographic protocols to ensure robustness and reliability.
- **Responsible & Ethical AI Security:** Ensure privacy preservation, fairness, and ethical data usage in AI-driven cryptographic systems, aligning with global security and data protection standards.

Suggested Tools:

- QuNetSim
- SimulaQron
- PennyLane
- GoogleColab

Applications:

1. Secure communication systems using Quantum Key Distribution (QKD)
2. Cybersecurity and quantum-safe encryption for AI systems
3. Financial transaction security and blockchain technologies
4. Defense and secure communication networks
5. Privacy-preserving AI systems and secure data sharing
6. Cloud security using quantum-resistant cryptographic methods
7. Secure IoT and smart city infrastructures

Mini Project Component

Industry-Driven Projects (Cyber security / Quantum Security Style)

These projects simulate real-world **secure communication and quantum cryptography applications**.

Project 1 – Quantum Key Distribution Simulator (BB84)

- **Problem:** Design and simulate the BB84 protocol for secure key exchange
- **Concepts used:** QKD, qubits, measurement, eavesdropping detection
- **Industry relevance:** Secure communication systems

Project 2 – Secure Communication System using QKD

- **Problem:** Develop a system combining classical encryption with quantum key distribution
- **Concepts used:** Hybrid cryptography, key exchange, encryption
- **Industry relevance:** Banking, defense, secure networks

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain classical and quantum cryptography concepts	L2
2	Apply information theory concepts in quantum systems	L3
3	Analyze quantum key distribution and key distillation techniques	L4
4	Evaluate quantum cryptographic protocols and their implementations	L5
5	Assess security aspects of quantum key distribution systems	L5

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2										2	
CO2	3	3	2	2	2							3	
CO3	2	3	2	3	2							3	2
CO4	2	3	2	3	3							3	2
CO5	3	3	3	3	3	2		2			2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Gilles Van Assche "**QUANTUM CRYPTOGRAPHY AND SECRET-KEY DISTILLATION**", Cambridge University Press

REFERENCE BOOKS:

2. **Post-Quantum Cryptography** – Springer.
3. **An Introduction to Quantum Computing** – Oxford University Press.
4. **Introduction to Quantum Information Science** – Oxford University Press.

E-RESOURCES:

1. https://qiskit.org
2. https://pennylane.ai

3. https://quantum.ibm.com
4. https://nptel.ac.in

Activity Based Learning (Suggested Activities in Class)

Activity 1: Quantum Protocol Simulation
Activity 2: Cryptographic Vulnerability
Analysis
Activity 3:Role-play demonstration of sender–receiver–eavesdropper scenario in secure communication
Activity 4: Group discussion on classical vs quantum cryptography and post-quantum challenges

****

Digital Business

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:	**Credits**:	03
Hours / Week	: 03 Hours	**Total Hours**:	45 Hrs L
L–T–P-J	: 3–0–0-0		

Course Vision:

The course aims to develop a strong foundation in digital business, e-commerce, and technology-driven business models for understanding modern digital economies and intelligent business systems. It enables students to analyze and manage digital platforms, online transactions, business analytics, and AI-driven decision-making processes in real-world applications. The course also fosters innovation, strategic thinking, and problem-solving skills required for designing efficient and sustainable digital business solutions in rapidly evolving technological environments.

Course Learning Objectives:

This Course will enable students to:

1. To understand the fundamental concepts of digital business, e-commerce, and their role in the modern digital economy.
2. To analyze various digital business models, including mobile commerce, social commerce, and platform-based ecosystems.
3. To apply digital business processes, payment systems, and logistics in real-world business scenarios.
4. To evaluate digital business applications such as B2B, B2C, FinTech, and e-governance systems.
5. To develop data-driven decision-making skills using business analytics, artificial intelligence, and modern digital tools.

Broad Course Description

This course provides a comprehensive understanding of digital business and how technology is transforming modern organizations and markets. It covers the fundamentals of e-commerce, mobile and social commerce, and the digital business ecosystem, including online transactions, logistics, and digital payment systems. The course further explores various digital business applications such as B2B, B2C, FinTech, and e-governance, highlighting their real-world impact. It also introduces business analytics, artificial intelligence, and data-driven decision-making to support strategic business operations. Overall, the course equips students with the knowledge and skills required to design, analyze, and manage digital business solutions in today's technology-driven environment.

UNIT – I	09 Hours

The Digital Revolution and Society, Digital Economy and Digital Enterprise, Virtual Communities and E-Commerce Definition, E-Commerce Frameworks, E-Business, E-Markets, Networks Classification of E-Commerce (by Nature of Transaction), E-Commerce Business Models (B2B, B2C, C2C, etc.), Drivers, Benefits, and Limitations of E-Commerce, Impact of E-Commerce on Business, Government, and Society.

UNIT – II	09 Hours

Data Collection and Tools

M-Commerce Applications and Benefits, Mobile Marketing and Location-Based Services
Social Commerce Concepts and Models, Social Business Networks and Enterprise Social Media, Social Media Marketing and Enterprise 2.0, Social Collaboration and Digital Engagement.

UNIT – III	09 Hours

E-Commerce Mechanisms and Online Purchasing Process, E-Marketplaces and Digital Intermediation Customer Shopping Mechanisms and Merchant Solutions, Auction Mechanisms and Dynamic Pricing Order Fulfilment Process and Logistics, Outsourcing, Partnerships, and Mass Customization, Digital Payments: Smart Cards, Stored Value Cards, Payment Gateways.	
UNIT – IV	**09 Hours**
B2B Applications: E-Marketplaces, E-Catalogs, Search Engines, E-Procurement and Supply Chain Integration, B2C Applications: Electronic Retailing (E-Tailing), Business Models and Revenue Models	
UNIT – V	**09 Hours**
Introduction to Business Analytics, Evolution and Need for Analytics in Digital Business, Types of Analytics: Descriptive, Predictive, Prescriptive, Tools for Analytics and Data-driven Decision	

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

1. **AI-enabled Digital Market Analysis:** Use AI tools to analyze market trends, customer behaviour, and digital demand patterns while ensuring ethical data usage and minimizing bias in insights.
2. **Intelligent E-Business Modelling:** Apply AI-based analytics to design and evaluate digital business models (B2B, B2C, platforms) with transparent and explainable assumptions.
3. **AI-driven Customer Engagement:** Utilize AI for personalization, recommendation systems, and social commerce strategies while maintaining privacy and responsible data handling.
4. **Smart Digital Operations:** Integrate AI in logistics, supply chain, and digital payment systems to improve efficiency, security, and scalability in business processes.
5. **Responsible AI & Digital Trust:** Evaluate ethical, legal, and societal implications of AI in digital business, ensuring fairness, accountability, transparency, and sustainability in decision-making.

Suggested Tools:

- HubSpot
- Power BI
- Uizard
- Miro
- Notion AI

Applications:

Market Intelligence & Trend Analysis:

Using tools like Google Trends and SEMrush to identify trending products, analyze competitors, and forecast market demand.

Customer Segmentation & Personalization:

Applying HubSpot and Google Analytics to segment customers based on behaviour and deliver personalized recommendations and targeted marketing.

Business Model Design & Strategy Optimization:

Using Canvanizer and Miro to design, visualize, and refine digital business models and strategies.

E-Commerce Development & Digital Transactions:

Leveraging Shopify and Razorpay to build online stores, manage transactions, and ensure secure digital payments.

Social Media Marketing & Customer Engagement:

Using Hootsuite and Mailchimp to manage campaigns, automate communication, and improve customer interaction.

Project Component

1. Identify a trending product/service and analyze its demand, target audience, and competitors.
2. Design and simulate a social media marketing campaign for a startup product.
3. Analyze customer journey and improve user experience using feedback and analytics.
4. Build a dashboard to analyze business data and generate insights using AI.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain the fundamentals of digital business, e-commerce models, and their impact on business and society.	L2
CO2	Analyze mobile commerce, social commerce, and IoT applications in the digital ecosystem.	L4
CO3	Apply concepts of digital business ecosystems, online processes, logistics, and digital payment systems.	L3
CO4	Evaluate various digital business applications including B2B, B2C, FinTech, and digital government services.	L5
CO5	Apply business analytics, AI, and data-driven decision-making techniques for digital business solutions.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	2	2	2	2	3			2	2	2	2		3
CO2	2	2	2	2	3			2	2	2	2		3
CO3	2	2	2	2	3			2	2	2	2		3
CO4	2	2	2	3	3			2	2	2	2		3
CO5	2	2	2	2	3			2	2	2	2		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Dave Chaffey, Tanya Hemphill, David Edmundson-Bird, Digital Business and E-Commerce Management, 7th Edition, Pearson Education, 2019.

REFERENCE BOOKS:

1. Efraim Turban, David King, Dennis Viehland, Jae Lee, Electronic Commerce – A

Managerial Perspective, 8th Edition, Pearson Education, 2016.

2. Judy Strauss, Adel Ansary, Raymond Frost (2017), E-Marketing, Prentice Hall.

3. Dave Chaffey, Fiona Ellis-Chadwick, Kevin Johnston, Richard Mayer (2018), Internet Marketing, Pearson Education.

E-RESOURCES:

1. https://onlinecourses.nptel.ac.in/noc25_mg152/preview
2. https://www.cognizant.com/us/en/glossary/digital-business
3. https://www.coursera.org/specializations/digital-business
4. https://www.udemy.com/course/professional-diploma-in-digital-business-development/?srsltid=AfmBOopmojTc_It09-JMMrCe4ORPytSUUulRPVkQ4pd3YgzEf7Pi0FE_&couponCode=CM260417IN

Activity Based Learning (Suggested Activities in Class)

1. Design a digital business model for a startup idea using real-world case studies.
2. Analyze an existing e-commerce platform (UI/UX, payment flow, customer journey) and present findings.
3. Create and present a mock digital marketing campaign for a product using content, targeting, and engagement strategies.

****

Internship

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:		**Credits**	:	02
Hours / Week	:	04 Hours	**Total Hours**	:	30 Hours J
L–T–P-J	:	0-0-0-4			

Course Vision: To provide students with practical industry exposure and professional experience for applying engineering knowledge, problem-solving skills, and modern technologies to real-world industrial and research challenges while enhancing employability, teamwork, and ethical practices.

Prerequisites: Completion of Year II (Core Theory: DSA, DAA, LDCO)

Course Learning Objectives:

This Course will enable students to:

1. To expose students to the industrial environment.
2. To create competent professionals for the industry.
3. To provide possible opportunities to learn, understand, and sharpen the real-time technical
    a. /managerial skills required at the job.
4. To work on a problem assigned by a mentor at industry, prepare an action plan and complete it within the time limit.
5. To learn, create/prepare a report for Project/research as used in industry in a productive and efficient way.
6. To strengthen industry-institute linkage and increase the employability of the students.

Broad Course Description: The Internship course provides students with hands-on industrial experience through supervised training in industry, research organizations, startups, or professional environments. The course enables students to apply theoretical concepts learned during the program to solve real-time technical and managerial problems using modern tools and technologies. Students work individually under the guidance of industry mentors and faculty supervisors to plan, execute, and complete assigned tasks within a defined timeline. The internship emphasizes professional ethics, communication skills, project planning, documentation, and industry-oriented problem-solving while strengthening industry–institute interaction and improving career readiness. Students are required to submit a detailed internship report, completion certificate, and progress documentation demonstrating the knowledge and skills gained during the training period.

Guidelines for Internship

The course includes 16 weeks of on-the-job training on current, industry-relevant problems through a supervised self-learning approach. The internship is an individual activity. The student should obtain approval from the chairman/supervisor to pursue. A student shall submit a brief proposal about the work to be carried out in the internship to a coordinator within 3 weeks after starting the internship.

A comprehensive report must be prepared and submitted to the department at the end of the semester. A certificate shall be attached to this report, duly signed by the competent authority of the industry for the successful completion of the internship. An attendance report shall also be attached to this report. The CIA evaluation will be done by a faculty mentor or Industry Supervisor. There is no SEE Exam for this course.

AI Integration & Authentic Intelligence: Reflection (AI² Pedagogy)

- Algorithmic thinking for AI model optimization
- Dynamic programming and greedy strategies in reinforcement learning
- Graph algorithms in neural network and social network connectivity
- Heuristic search for game and AI decision systems

- Responsible algorithmic design (fairness, bias minimization)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Understand the modern tools used in the field of Computer science and engineering for product development.	L2
CO2	Demonstrate ethical conduct and professional accountability while working in a team for the benefit of society.	L2
CO3	Understand the resources requirement and planning to facilitate the project success.	L3
CO4	Develop and refine oral and written communication skills.	L3
CO5	Demonstrate knowledge of the industry in which the internship is done.	L3

Table: Mapping Levels of COs to POs / PSOs

COS	Program Outcomes (POs)											Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	3	3	3	3	3	3	3	3	3	3	3	3
CO2	3	3	3	3	3	3	3	3	3	3	3	3	3
CO3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO4	3	3	3	3	3	3	3	3	3	3	3	3	3
CO5	3	3	3	3	3	3	3	3	3	3	3	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

****

Capstone Project - Phase II

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VIII

Subject Code	:	**Credits**	: 10
Hours / Week	: 03 Hours	**Total Hours**	: 150 Hours J
L–T–P–J	: 0–0–0–20		

Course Vision:

The course aims to develop students' ability to design, implement, test, and deploy real-world engineering solutions using modern software engineering practices and tools. It enhances technical, analytical, research, and project management skills through collaborative project development and problem-solving. The course also encourages innovation, teamwork, technical communication, and research contributions through publications, patents, hackathons, and project exhibitions. It prepares students to handle professional engineering projects with effective documentation, presentation, and evaluation of results.

Course Learning Objectives:

This Course will enable students to:

1. Detailed design of the solution to the problem statement and project management using software engineering skills.
2. Write efficient code and test the code to find any bugs and resolve the same leading to completion and deployment of the project using modern tools.
3. Analyze and synthesize the project results.
4. Demonstrate knowledge and understanding of writing the publication/report.
5. Able to work in teams and present the project work.

Broad Course Description:

1. The problem statement selected in Major Project Phase-I (VII semester) will be carried in the VIII semester.
2. Phase-II comprises of the detailed design, implementation, and testing results during the internal and external review.
3. Each Project team needs to submit the technical paper or patent or participate in hackathons and project exhibitions as well as apply for various state and national funding agencies within the stipulated time frame by the university.
4. There will be CIA evaluation (Project reviews) done by a committee of senior faculty of the Department.
5. Additionally, there will be a Semester End Evaluation of the work done that would include an internal Faculty and an External Academic Expert.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Conduct a survey of several available literature in the preferred field of study to find the recent advances and gaps.	L4
2	Implement the mathematics concept and engineering fundamentals, and specialization to design a solution using modern tools for the defined problem.	L6
3	Experimenting and evaluating the results from test data to provide a conclusion to the project work	L5
4	Demonstrate an ability to work in teams and to prepare quality documents of project work & exhibit technical presentation skills	L6

Table: Mapping Levels of COs to POs / PSOs													
COS	**Program Outcomes (POs)**											**Program Specific Outcomes (PSOs)**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2
CO1	3	3	3	3	3	3	3	3	3	3	3	3	3
CO2	3	3	3	3	3	3	3	3	3	3	3	3	3
CO3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO4	3	3	3	3	3	3	3	3	3	3	3	3	3

****