SCHOOL OF ENGINEERING

Department of

Computer Science & Medical Engineering

SYLLABUS 1ST YEAR

B.Tech. PROGRAMME– 2025-29 BATCH

Academic Year 2025-26

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

LINEAR ALGEBRA & DIFFERENTIAL EQUATIONS

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

SEMESTER – I

Course Code	:	25ENG1101	**Credits**	:	04
Hours / Week	:	04 Hours	**Total Hours**	:	60 Hours
L–T–P–J	:	3–1–0–0			

Course Learning Objectives:

This Course will enable students to:

1. **Understand** basic concepts of linear algebra to illustrate its power and utility through applications to science and Engineering.
2. **Apply** the concepts of vector spaces, linear transformations, matrices and inner product spaces in engineering.
3. **Explain** the course algebraically as well as geometrically.
4. **Solve** problems in cryptography, computer graphics and wavelet transforms.

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** means it includes not only traditional lecture method, but different type 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 rather than simply recall it.
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	**12 Hours**
INTRODUCTION: System of Linear equations, Row reduction and echelon form, Rank of a matrix by row echelon form, Gauss elimination, Inverse of a matrix by Gauss Jordan LU.	
UNIT – II	**12 Hours**
Vector spaces – Subspaces, Linear independence – Span - Bases and Dimensions -Finite dimensional vector spaces, Dimensions, finite dimensional vector spaces.	
UNIT – III	**12 Hours**
Linear transformation - Matrices of linear transformations, Vector space of linear transformations – Inner Product, Orthogonal Vectors – Projections, Gram- Schmidt Orthogonalization process.	
UNIT – IV	**12 Hours**
Introduction to Eigenvalues and Eigenvectors. Diagonalization of a Matrix.	

UNIT – V	12 Hours
Linear second order ordinary differential equation with constant coefficients, Solutions of homogenous and non-homogenous equations, Method of variation of parameters, Solutions of Cauchy-Euler and Cauchy-Legendre differential equations	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Apply the abstract concepts of matrices and system of linear equations using decomposition methods.	L2
CO2	Explain the basic notion of vector spaces and subspaces.	L3
CO3	Apply the concept of vector spaces using linear transforms which is used in computer graphics and inner product spaces	L2
CO4	Implement the concept of linear transforms and orthogonality in computer graphics and imaging.	L3
C05	Make use of techniques for solving differential equation in dynamical systems.	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	3
CO1	3	3	2	2	2	-	-	-	-	1	2	2	1	-
CO2	3	3	2	2	2	-	-	-	-	1	2	2	1	-
CO3	3	3	3	2	3	-	-	-	-	1	2	3	2	-
CO4	3	3	3	3	3	-	-	-	-	1	2	2	3	-
CO5	3	3	2	3	2	-	-	-	-	1	3	2	1	-

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

TEXT BOOKS:

1. D C Lay, S R Lay and JJ McDonald, Linear Algebra and its Applications, Pearson India, Fifth edition, 2016.
2. Gilbert Strang, Linear Algebra and its Applications, 4th Edition, Thomson Brooks/Cole, Second Indian Reprint 2007.
3. Bernard Kolman and David, R. Hill, Introductory Linear Algebra- An applied first course, 9th Edition, Pearson Education, 2011.
4. George B. Thomas, D. Weir and J. Hass, Thomas' Calculus, 13th edition, Pearson, 2014.
5. Erwin Kreyszig, Advanced engineering mathematics, Wiley, London, 1972.

REFERENCE BOOKS:

1. Gilbert Strang, Introduction to Linear Algebra, 5th Edition, Cengage Learning, 2015.
2. B S Grewal, Higher Engineering Mathematics, 42nd Edition, Khanna Publishers.
3. Stephen Andrilli and David Hecker, Elementary Linear Algebra, 5th Edition, Academic Press 2016.
4. Robert C Busby, Contemporary linear algebra, Howard Anton, Wiley, 2003.

5. Farin and Hansford, Practical Linear Algebra, CRC Press, 2013.

E-Resources:

1. https://nptel.ac.in/courses/111101115
2. https://nptel.ac.in/courses/111108066
3. Linear Algebra Basics | Coursera
4. https://nptel.ac.in/courses/111108081
5. https://nptel.ac.in/courses/111106100
6. Differential Equations for Engineers Course (HKUST) | Coursera

Activity Based Learning (Suggested Activities in Class)

1. Introduce the concept of matrix transformations, such as translation, rotation, scaling, and reflection. Provide visual examples and interactive tools that allow students to manipulate shapes and observe the effects of different transformation matrices.
2. Using real-life scenarios or word problems to make the activity of solving linear equations using matrix method.
3. Some real-world scenarios that can be modelled using ODEs, such as population growth, radioactive decay, or chemical reactions that can be discussed and solve using appropriate methods.
4. Real world problem solving and puzzles using group discussion.
5. Demonstration of solution to a problem through experiential learning.

****

ENGINEERING GRAPHICS & DESIGN THINKING

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I/II

Course Code	: 25ENG1102	**Credits**	:	04
Hours / Week	: 05 Hours	**Total Hours**	:	60 Hours
L–T–P–S	: 2–1–2–0			

Course Learning Objectives:

This Course will enable students to:

1. Create awareness and emphasize the need for Engineering Graphics & design thinking through Manual Sketching & AutoCAD Software
2. Learn using professional CAD software for construction of geometry
3. Understand the concepts of orthographic and isometric projections
4. Draw orthographic projection of lines and solids by Manual Sketching & AutoCAD Software
5. Draw development of surfaces of solids & isometric projections of solids
6. Create simple engineering 3D components using design thinking methods
7. Work in a team for creating conceptual design of products
8. Learn application of CAD techniques for Innovation methods and tools on real world problem through fusion 360 Software

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** means it includes not only traditional lecture method, but different type 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 rather than simply recall it.
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	**6 Hours**

Fundamentals of Engineering Graphics & Projections
(Manual & Computer Sketching)

Principles of Engineering Graphics and their significance, lettering & dimensioning, Scaling, Principle of Orthographic Projections, Introduction to AutoCAD, Basic commands. Projection of points, Projection of straight lines inclined to both the principal planes
(First Angle Projection only and Using AutoCAD Software)
(For CIA only-Continuous Internal Assessment)

UNIT – II	6 Hours
Projection of Solids & Development of Surfaces **(Manual & Computer Sketching)** Classification of solids, Projection of regular solids like prisms, pyramids, cylinder & cone inclined to both the planes (change of position method, First Angle Projection only and Using AutoCAD Software) (**For CIA only)** Development of lateral surfaces of regular solids – Prisms, pyramids cylinders and cones (Using AutoCAD Software) (**For SEE Only- Semester End Exam)**	
UNIT – III	**6 Hours**
Isometric Projections **(Manual & Computer Sketching)** Principles of Isometric Projection, Isometric Scale, Isometric View, Isometric projection of combination of two solids. Conversion of Isometric Views to Orthographic Views & vice versa (Using Auto CAD software) **(For SEE Only)**	
UNIT – IV	**6 Hours**
Introduction to Design Thinking & Innovation (Mini Project-Group) Empathize and Define, Ideate and Create, Prototype and Test, Iterate and Refine, Product development cycle, Use of Diagrams and Maps in Design Thinking, Storytelling, Innovation- aesthetics, usability, and product appearance, Creativity and Idea Generation, solving problems, Value creation, Exercises and case-based discussions (**For CIA only)**	
UNIT – V	**6 Hours**
CAD techniques for Design Thinking & Innovation **(Computer Sketching)** Introduction to CAD & engineering design, Top-down design, bottom-up design, create 3D models and technical drawings, Multiview drawings, Modelling with Autodesk Fusion-Part objects & assembly drawings (Using Fusion 360 software) **(For SEE Only)**	
LAB COMPONENT	
Note: Problems to be solved in first quadrant system. Manual & Computer Sketching problems in sketch book and also take print out of the problems. Following are practical/laboratory experiments to be carried out: 1. Usage of various commands in AutoCAD software and few exercises on the above commands 2. Practice Problems on Projections of Lines and Solids using Manual Sketching & AutoCAD Software 3. Solve Problems on Development of Surfaces using AutoCAD software 4. Practice problems on Isometric Projections -Combination of solids 5. Solve Isometric Projections to orthographic Projections and vice versa problems using AutoCAD software 6. Mini Project Group work on Introduction to Design Thinking for Innovations 7. 3D models and technical drawings 8. Modelling with Autodesk Fusion-Simple Part objects & assembly drawings	

Department of Computer Science & Medical Engineering

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Make use of instruments, dimensioning & tolerance principles, conventions and standards related engineering drawing	L2
CO2	Construct orthographic projections of lines and solids	L3
CO3	Develop lateral surfaces of solids and construct isometric projections of solids	L2
CO4	Apply the design thinking principles for innovative product development	L3
C05	Make use of AutoCAD & Fusion 360 software for modelling engineering components	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	3
CO1	3	2	2	-	2	-	-	-	-	-	1	1	2	-
CO2	3	2	3	1	2	-	-	-	-	-	1	1	2	-
CO3	3	2	3	1	3	-	-	-	-	-	1	1	3	-
CO4	2	2	3	2	2	1	1	2	1	1	2	1	2	3
CO5	3	2	3	2	3	-	-	-	-	-	2	1	3	-

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

TEXT BOOKS:

1. Engineering Drawing, Bhatt N.D., 54th Edition, Charotar Publishing House, Gujarat,
   i. India, 2023
2. Engineering Drawing & Graphics+Autocad, K Venugopal, Fifth Edition, New Age International Publishers, 2011.
3. Engineering Design- A Project Based Introduction, C. L. Dym and Patrick Little, John Wiley & Sons, 2022 Andreas Gebhardt and Jan-Steffen Hötter "Additive Manufacturing: 3D Printing for Prototyping and Manufacturing", Hanser publications, United States, 2015, ISBN: 978- 1 - 56990-582-1.
4. Design Thinking: A Comprehensive Textbook, Shalini Rahul Tiwari, Rohit Rajendra Swarup, Wiley,2024

REFERENCE BOOKS:

1. A Textbook of Computer Aided Engineering Drawing, Gopalakrishna, K. R. and Sudheer Gopala Krishna, Subash Publishers, Bangalore, India, 2017
2. Engineering Drawing with Introduction to AutoCAD, Dhananjay. A. J, Tata McGraw-
   i. Hill Publishing Company Ltd, 2018
3. Product Design and Development, Karl T Ulrich, Steven D Eppinger, Seventh Edition,
   i. McGraw-Hill Education, 2020

E-Resources:

1. https://archive.nptel.ac.in/courses/112/102/112102304/
2. https://nptel.ac.in/courses/112103019
3. https://fractory.com/engineering-drawing-basics/
4. https://www.tutorialspoint.com/design_thinking/index.htm
5. https://archive.nptel.ac.in/courses/107/101/107101086/

Activity-Based Learning (Suggested Activities in Class)

1. Activity which makes students to apply the concepts learned in the course to the practical engineering graphics will be discussed in class.
2. Activity provides space to students giving responsibility for their own design & engineering drawing methods for the products
3. Activity that makes the students for the development of skill set in computer drafting
4. Activity that makes the students to have critical thinking, developing a mindset, problem-solving and teamwork in design thinking process.
5. Real world problem solving and puzzles using group discussion.
6. Demonstration of solution to a problem through experiential learning.

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

CHEMISTRY FOR CSME

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

SEMESTER – I/II

Subject Code	: 25ENG1106	**Credits**	: 04
Hours / Week	: 05 Hours	**Total Hours**	: 60 Hours
L–T–P–S	: 2–1–2–0		

Course Learning Objectives:

1. To enable students to acquire knowledge on the basics of biochemistry for medical engineering applications.
2. To develop an intuitive understanding of bio chemistry by emphasizing the related branches of engineering.
3. To provide students with a solid foundation in modelling of molecules.
4. To understand the Principles of Drug Design and scoring function of small molecules.

Teaching-Learning Process (General Instructions):

These are some of the innovative pedagogical approaches to accelerate the attainment of the various course outcomes.

1. **Lecture method**: Chalk and talk method, and demonstrations using smart boards may be adopted to achieve the course outcomes.
2. **Interactive Teaching: Active learning** that includes brainstorming, group work, formulating questions, notetaking, and annotating.
3. Show **Video**s to explain and illustrate the various concepts.
4. Encourage **Collaborative** learning in the class.
5. **Problem Based Learning**, may foster students' analytical skills, ability to evaluate, and process the information.
6. Inculcate a culture of research and encourage students to express their own creativity.

Unit I	**Carbohydrates**	**06 Hours**

Monosaccharides - structure of aldoses and ketoses, ring structure of sugars, conformations of sugars, mutarotation, anomers, epimers and enantiomers, structure of biologically important sugar derivatives, oxidation of sugars. Formation of disaccharides, reducing and non- reducing disaccharides. Polysaccharides – homo- and heteropoly saccharides, structural and storage polysaccharides. Structure and role of proteoglycans, glycoproteins and glycolipids (gangliosides and lipopolysaccharides). (definition, structure, functions, examples only) Carbohydrates as informational molecules, working with carbohydrates (applications of carbohydrate)

Unit II	**Amino acids, peptides, proteins and Nucleic acids**	**06 Hours**

Classification, chemical and physical properties of amino acids. Peptide bond: planarity and dihedral angles, Ramachandran plot. Structural hierarchy of proteins: primary, secondary, super-secondary, tertiary and quaternary structures. Classification of proteins, properties: isoelectric point, zwitterions, and precipitation reactions. Globular and fibrous proteins: Classification, chemical and physical properties.

Fatty acids: saturated, unsaturated and essential fatty acids; rancidity, saponification number, iodine number, acid number and Reichert–Meissel number.

Chemistry of purine and pyrimidine, nucleosides and nucleotides. Types of DNA: structure and properties of A-, B- and Z-DNA. Denaturation, renaturation.

Nucleotides - structure and properties. Nucleic acid structure – Watson-Crick model of DNA. Structure of major species of RNA - mRNA, tRNA and rRNA. Nucleic acid chemistry - UV absorption, effect of acid and alkali on DNA. Other functions of nucleotides - source of energy, component of coenzymes, second messengers (examples & functions only

Unit III	**Introduction to molecular modeling**	**06 Hours**

Cambridge Structural Database (CSD), Protein Data Bank (PDB), File format and information in PDB and CSD databases; Molecular Modelling: Molecular Graphics, 3-D Models of Organics and Biomolecules, Potential Energy Surfaces.

Case Study: Visualization of 3D Models with Gauss View/Avogadro/Chemcraft

Unit IV	Introduction to drug design and delivery	06 Hours
The Principles of Drug Design: The Drug Discovery Pipeline and Costs of Drug Discovery, Lipinski's Rules of 5, Drug Metabolism, Toxicity and Side Effects, ADMET (Absorption, Distribution, Metabolism, Excretion, Toxicity) Models, Drug resistance and promiscuity, Drug Delivery systems, Pharmacophore Modelling and alignment, Lead Optimization, Ligand-Based and Structure-Based Design; Molecular Docking, Introduction, Scoring Functions, Applications of Docking; Case Study: Setting up a Molecular Docking for a simple biomolecule.		
Unit V	**Application of AI/ML in Drug and Materials Design**	**06 Hours**
Application of AI/ML in Drug and Materials Design: Introduction to Linear Free Energy Relationships, Structure-Activity Relationships and the Similarity Principle, Virtual High-throughput Screening, Feature selection – Genetic Algorithms, Model Validation, Case Study: Modeling with random forests, PLS, SVM and Neural Networks.		

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Identify the terms and principles of Biochemistry involved in Medical Engineering and Technology	L3
2	Explain the phenomenon of Biochemistry to describe the methods of Synthesis and applications in daily life	L3
3	Solve the problems of Biochemistry that are pertinent in socio-economic and environmental issues.	L3
4	Apply the basic concepts of Biochemistry in quantification of industrially relevant parameters.	L4
5	Analyze the structure, properties and processes associated with chemical substances in multidisciplinary situations	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	3
CO1	3	3	2	1	1	-	2	-	-	-	1	3	1	2
CO2	3	3	2	1	2	-	2	-	-	-	1	3	1	2
CO3	3	3	2	2	1	1	2	-	-	-	1	3	1	2
CO4	3	3	3	2	3	-	1	-	-	-	1	3	2	2
CO5	2	3	3	2	2	1	2	-	-	-	2	3	2	3

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

Activity Based Learning (Suggested Activities in Class)

1. Analyze research problems by reading research articles, group discussion, and presentations.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects, taking students on an educational tour.

Text books:

1. Lehninger: Principles of Biochemistry (2013) 6th ed., Nelson, D.L. and Cox, M.M.,
2. W.H. Freeman and Company (New York), ISBN: 13: 978-1-4641-0962-1 / ISBN: 10:1-4292- 3414-8.
3. Biochemistry (2012) 7th ed., Berg, J.M., Tymoczko, J.L. and Stryer L., W.H. Freeman and Company (New York), ISBN:10:1-4292-2936-5, ISBN:13:978-1-4292-2936-4
4. Fundamental of Biochemistry, Voet and Voet- provide necessary details on latest edition Edited by Prof. Hiren K Das (JNU)

Web links and Video Lectures (e-Resources):

1. http://libgen.rs/

CHEMISTRY- LABORATORY **Total: 26 Hrs**

1. 3D Visualization and Structural Analysis of Carbohydrates Using Avogadro
2. Exploration of Protein Structures Using the Protein Data Bank (PDB) and PyMOL
3. Analysis and Comparison of Molecular File Formats: PDB, MOL, and CIF
4. Molecular Docking of Aspirin with COX Enzyme Using PyRx
5. Drug-Likeness Evaluation Using Lipinski's Rule of Five via SwissADME
6. In Silico Prediction of ADMET Properties Using Online Tools
7. Development of a QSAR Model for Biological Activity Prediction Using Machine Learning
8. Similarity Search and Virtual Screening of Drug Analogs Using PubChem and ChemMine
9. Qualitative Tests for Carbohydrates
10. Isolation of Genomic DNA from Plant Cells
11. Estimation of Protein by Lowry's Method

Reference Books

1. Dayananda Sagar University laboratory manual.
2. J. Evanegeline, Manual of Practical Medical Biochemistry, Jaypee Brothers Medical Publishers (P) Ltd.

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

ENGINEERING SCIENCE COURSE – I FOR COMPUTER SCIENCE CLUSTER
OBJECT ORIENTED PROGRAMMING

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

SEMESTER – I/II

Course Code	:	25ENG1107	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–J	:	3–0–0–0			

Course Learning Objectives:

This Course will enable students to:

1. **Understand** different programming paradigms, significance of object-oriented programming approach and their applications.
2. **Make use of** Python programming environment to develop programs using conditionals, iterations, functions, strings and files to store and retrieve data in system.
3. **Gain** skills to develop python programs using core data structures like Lists, Tuples, Sets and Dictionaries.
4. **Describe** the concepts of object-oriented concept using class, objects, methods. Polymorphism and different levels of inheritance.
5. **Explain** single and multiple exception handling capabilities in python.

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*** means it includes not only traditional lecture method, but different *type 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 rather than simply recall it.
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

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING AND PYTHON

Object oriented programming features, Applications, Features of Python, History of Python, variables and identifiers, Data types, input operation, comments, Reserved words, Indentation, Operators and Expressions, Type conversions.

(Text Book-1: Chapter 2: 2.4,2.6 Chapter 3: 3.1,3.2,3.6,3.7,3.8,3.9,3.10,3.11,3. 12,3.16)

DECISION AND LOOP CONTROL STATEMENTS:

Conditional branch statements, Iterative statements, Nested loops, break, continue, pass, The else statement used with loops.

(Text Book-1: Chapter 4: 4.1-4.8)

UNIT – II — 09 Hours

FUNCTIONS:

Need for functions, Function definition, Function call, Scope, Return statement, Lambda functions, Recursive functions.

(Chapter 5: 5.1 to 5.7, 5.10)

PYTHON STRINGS: String operations, Immutable, string formatting operator, built-in string methods, string slices, membership operator, comparing strings, Iterating strings. ***(Chapter 6: 6.1 to 6.9)***	
UNIT – III	**09 Hours**
DATA STRUCTURES IN PYTHON: **LIST:** Introduction, create list, Access values in List, update list, Nested lists, Cloning lists, List operations, List methods, updating values in list, list comprehensions, Looping in lists. **TUPLE:** Introduction, create tuple, Access values in tuple, update tuple, delete elements in tuple, Tuple operations, Tuple assignment, tuples return multiple values, nested tuples, tuple methods, Advantage of tuple over lists. **SETS:** Create sets, set operations, update a set, Iterate a set. **DICTIONARIES:** Create dictionary, Add and update item in a dictionary, delete items, sort items, loop over a dictionary, nested dictionaries, dictionary function and methods, difference between list, tuple set and dictionaries. ***(Chapter 8: 8.2, 8.4, 8.5, 8.6)***	
UNIT – IV	**09 Hours**
FILE HANDLING METHODS: File path, File types, File operations, Rename and delete files. ***(Chapter 7: 7.1 to 7.5, 7.7)*** **USER DEFINED CLASSES & OBJECTS:** Classes, Objects, class method and self- Argument, constructor, destructor, class variables, public and private data members, private methods, Calling methods. ***(Chapter 9: 9.1 to 9.10)***	
UNIT – V	**09 Hours**
INHERITANCE: Introduction, inherit classes in python, Polymorphism and method overriding, Types of inheritance ***(Text Book: Chapter 10: 10.1 to 10.3)*** **ERROR AND EXCEPTION HANDLING:** Errors, Handling exceptions, Multiple except blocks, Multiple exceptions, except block without exception, the else clause, finally block, clean- up action ***(Text Book: Chapter 12: 12.1 to 12.6, 12.10 to 12.12)***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Develop** a python program using four conditionals, definite loop, indefinite loop with jump statements.	L3
2	**Develop** a python program using lambda, recursive functions, strings and files to store and retrieve the data from the system	L3
3	**Develop** python programs using core data structures like Lists, Tuples, Sets and Dictionaries	L3
4	**Implement** the concepts of object-oriented concept using class, objects, methods, polymorphism and different levels of inheritance.	L4
5	**Implement** single and multiple exception handling program capability in python	L3

TEXT BOOKS:

1. Reema Thareja, "Python programming: Using problem solving approach", 2nd Edition, Oxford university press, 2019.

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**	**3**
CO1	3	3	3	2	3	-	-	-	-	-	1	3	2	-
CO2	3	3	3	2	3	-	-	-	-	-	1	3	2	-
CO3	3	3	3	2	3	-	-	-	-	-	1	3	2	-
CO4	3	3	3	3	3	-	1	1	-	-	2	3	3	1
CO5	3	3	2	2	3	-	1	-	-	-	2	2	2	1

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

REFERENCE BOOKS:

1. John V Guttag, "Introduction to Computation and Programming Using Python", The MIT press, 3rd edition, 2021.
2. Tony Gaddis, "Starting out with python", 4th edition, Pearson, 2019.
3. Allen Downey, Jeffrey Elkner and Chris Meyers, "How to think like a Computer Scientist, Learning with Python", Green Tea Press, 2014.
4. Richard L. Halterman, "Learning to Program with Python", 2011.
5. Charles Dierbach, "Computer Science Using Python: A Computational Problem-Solving Focus", John Wiley, 2012.

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through programming.

OBJECT ORIENTED PROGRAMMING EXPERIMENTS for Practice

List of Programming Experiments:

1. Python Program for Data Handling and Expression Evaluation.
2. Python Program for Quadratic Equation Roots and Number Analysis.
3. Python Program for Function Illustration and Module Creation.
4. Python Program for String Operations and Data Validation.
5. Python Program for File Handling and Script Copying.
6. Python Program for Data Structures and Built-in Methods.
7. Python Program for Object-Oriented Concepts and Inheritance.
8. Python Program for catching an error using try and except blocks.

****

TECHNICAL ENGLISH

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

SEMESTERS – I/II

Subject Code	: 25ENG1110	**Credits**	:	02
Hours / Week	: 02 Hours	**Total Hours**	:	30 Hours
L–T–P-J	: 2-0-0-0			

Prerequisites:

Developing Communicative competence: Enhancing the Language competence in the technical discourse and augmenting the strategic competence in the social and professional environment.

Course Learning Objectives:

This course will enable students to:

1. To enable students to improve their lexical and grammatical competence.
2. To enhance their verbal and nonverbal communication in a professional environment
3. To optimize oral and written communication.
4. To familiarize the students with employability and job search skills.
5. To enhance the students with soft skills
6. To inculcate critical thinking

Teaching-Learning Process (General Instructions):

These are some of the innovative pedagogical approaches to accelerate the attainment of the various course outcomes.

1. ***Lecture method***: Anecdotes, case studies and Examples from real-life situations may be adopted along with the traditional method of chalk and talk to achieve the course outcome.
2. ***Interactive Teaching: Active learning*** may be adopted which includes brainstorming, Teamwork, focused listening, formulating questions, note-taking, and Role play.
3. ***Collaborative*** learning through Debates and Group Discussion
4. ***Activity-based learning*** to inculcate Critical ***thinking*** – conceptualizing, applying, analyzing, synthesizing, and/or evaluating information from observation, perception, and expression. Minimum three higher-order questions from the real-world context
5. ***Problem-Solving method through*** Activities and discussion / Minimum of three situations to inculcate ***Problem-Solving skills*** and encourage the students to come up with creative ways to solve the problem
6. ***Audio-visual methods*** through language Lab in the teaching of LSRW skills.
7. ***Short films/ Ted talks/ Videos/Animation*** films to explain the functioning of various concepts.
8. ***Flipped learning***
9. ***Peer learning / Peer tutoring***

Module – I	**06 Hours**

Grammar and Usage, Language and Communication.

Branches of Grammar and Vocabulary-Difference between hearing and listening. Reading: Extensive and Intensive Speaking: Different aspects of Effective Speaking. Pronunciation Guidelines- - Common Errors of Pronunciation- Presentation Skills Various Techniques for Neutralization of Mother Tongue Influence

Objective:

1. Revising and practicing grammar will help students to optimize their language Competence
2. Listening steps up language learning and improves pronunciation
3. Reading provides exposure to the chosen field and helps in the coherence of the thought process
4. Speaking improves one's ability to construct phrases naturally and spontaneously in everyday discussions, Clarity and comprehensiveness in speech.

Communicating effectively in the Professional environment, to interact with the Peer group and to involve in collaborate initiatives.

Module – II	**06 Hours**

Research skills: The difference between area and topic, how to arrive at a topic (Critical thinking), Group Discussion: Comprehensive techniques, Organization and presentation, fluency and articulation of ideas

Objective:

1. GD helps individuals to achieve the skills of organizing and presenting the ideas and concepts in a cohesive manner and to overcome the inhibition of expression in communication
2. Enhancing problem solving skills, communicative techniques, general awareness and self confidence
3. Improving verbal and nonverbal communication and to use appropriate body language

Module – III	**06 Hours**

Technical Writing: Writing and Minutes of the Meeting, Memo and E-mail Etiquette, Designing PowerPoint slides, Referencing Skills for Academic Report Writing, Technical Paper

Objective:

1. Familiarizing with email etiquettes and correspondence provides learners to form an excellent first impression, establishing trust and confidence. Meeting minutes allows to access information such as facts, opinions, votes cast, conflicts, attendees, and other crucial elements at the workplace.
2. Technical writing techniques enable the knowledge in the relevant domain and creates better content based on the need of the target group
3. Following the Academic conventions helps the students to optimize their reference skills and use references to acknowledge the input of other authors and scholars in their work and avoid plagiarism
4. Writing technical reports develop competence in creating a legally bound account of efforts and choices and engineering technical report propose a solution to a problem in order to inspire action.

Module – IV	**06 Hours**

Drafting Curriculum Vitae, Resumes, and Cover- Letters.

Types of Resumes, Preparing Resume, CV and Cover- Letter. Difference between Resume & Curriculum Vitae, Interview Techniques and Etiquette.

Objective:

Learning the specifics of designing a CV or Resume helps in the effective presentation of their achievements and skills, and a cover letter is a chance for them to exhibit a few aspects of their personality.

Course Outcome	Description	Bloom's Taxonomy Level -
At the end of the course the student will be able to:		
1	Developing language competence **improves** one's ability to construct phrases naturally in everyday discussions and Communication skills and soft skills enhance the self-confidence of students	L3
2	**Applying** the fundamentals of technical writing techniques provides adequate exposure to the respective domain and creates better content. **Implementing** the technicalities of writing provides better exposure in the domain.	L3
3	**Following** an appropriate style of email reveals the aspect of professionalism. **Develop** technical writing skills to increase the quality of the work	L3
4	**Practicing** communication with greater clarity and ease enable the students to discuss a wide spectrum of topics.	L3
5	Writing resumes or curriculum vitae provides a practice to exhibit their	L3

	skills and achievements concisely and writing a covering letter enables them to express their personality in the formal context.	

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	3
CO1	1	1	1	-	-	-	-	2	3	-	2	-	-	2
CO2	1	2	2	1	1	-	-	-	3	-	2	1	-	2
CO3	1	2	2	1	1	-	1	-	3	-	2	1	-	2
CO4	1	1	1	-	-	1	1	2	3	-	2	-	-	2
CO5	1	1	1	-	-	-	-	1	3	1	2	-	-	2

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Day. R A. "Scientific English: A Guide for Scientists and Other Professional". 2nd Edition, Hyderabad: Universities Press, 2000. Print.
2. Ashraf Rizvi M. "Effective Technical Communication". McGraw Hill Education, 2017. Print.
3. Eastwood J. "Oxford Practice Grammar". Oxford University Press, 1999. Print.
4. Swan M and Walter C. "Oxford English Grammar Course". Oxford University Press, 2011. Print.
5. Dale, Carnegie. "The Quick and Easy Way to Effective Speaking". JAICO Publishing House, 2019. Print.
6. Chauhan, Gajendra S and Smita, Kashiramka. "Technical Communication". India: Cengage Learning India Private Limited, 2018. Print.
7. Bailey, Stephen. "Academic Writing: A Handbook for International Students". 5th Edition, Routledge, 2017. Print.
8. Kumar, Shiv K and Nagarajan, Hemalatha. "Learn Correct English: Grammar, Composition and Usage". 1st edition, India: Pearson, 2005. Print.
9. Board of Editors. Language and Life: A Skills Approach. Orient BlackSwan, 2018.
10. Sudharshana, NP and C Savitha. English for Engineers. Cambridge University Press, 2018.
11. Kumar, Sanjay and Pushp Lata. English Language and Communication Skills for Engineers. Oxford University Press, 2018.
12. Thomson, A.J. and Martinet, A.V. A Practical English Grammar, OUP, New Delhi: 1986
13. Anne Laws, —Writing Skills||, Orient Black Swan, Hyderabad, 2011
14. Richards, C. Jack. Interchange Students' Book-2 New Delhi: CUP, 2015.

E-Resources:

1. https://gnindia.dronacharya.info/ME/Common-Subjects/Downloads/Technical-Communication/Books/Technical-Communication-Book-9.pdf. Web
2. https://projects.iq.harvard.edu/files/hks-communications-program/files/ho_murphy_michael-pp-slides_9_30_14.pdf. Web.
3. https://www.youtube.com/watch?v=TR0JZiapxXM. Web.
4. file:///C:/Users/rochn/Downloads/ManualofEnglishGrammarandComposition_1001257.pdf
5. https://www.youtube.com/watch?v=f5Tao6KHV5w. Web.
6. https://www.sastra.edu/nptel/download/Prof%20GPRagini/pdf_New/Unit%2026.pdf. Web.

7. https://www.hansrajcollege.ac.in/hCPanel/uploads/elearning/elearning_documen t/English_ communication_chapter_13.pdf. Web.
8. https://www.youtube.com/watch?v=voyGGhlpBR8. Web.

Activity Based Learning (Suggested Activities in Class)

1. Observing and responding appropriately to the real-life situations.
2. Encouraging students to participate in Group discussions.
3. Articulating internal observations precisely and confidently through extempore.
4. Producing sentences easily without any grammatical errors in speaking, writing essays, and creative writing.
5. Conducting mock interviews, to refine their expressions, familiarize them with the interview techniques, and provide training for the spontaneous response to tricky questions.
6. Directing students for PowerPoint presentations and orienting them towards the higher order skills of expressing their ideas and concepts with cohesion.

****

ENVIRONMENTAL SCIENCE

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

SEMESTER – I/II

Course Code	: 25ENG1111	**Credits**	: 02
Hours / Week	: 02 Hours	**Total Hours**	: 30 Hours
L–T–P–S	: 2–0–0–0		

Course Learning Objectives:

1. To understand the evolution and diversity in nature.
2. To understand the concepts of environment, pollution, energy resources
3. To learn water as a resource, rain water harvesting as a method of conversation of water
4. To explain solid waste and its management
5. To learn environmental Protection Act laws, environmental Impact Analysis and air monitoring

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** means it includes not only traditional lecture method, but different type 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.
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 rather than simply recall it.
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	**06 Hours**

ECOSYSTEMS and BIODIVERSITY: Concept of an ecosystem. Structure and function of an ecosystem. Producers, consumers and decomposers. Energy flow in the ecosystem. Food chains, food webs and ecological pyramids.

ENVIRONMENT and POLLUTANTS: Definition of environment; Scope and importance of environmental studies; Basic concepts: Xenobiotic, natural & anthropogenic; why are we concerned? Eco-kinetic & Bio-kinetic Properties of a xenobiotic, Dose-Response Relationships; 3 T's, Chronic and acute health effects.

For CIA only

UNIT – II	**06 Hours**

POLLUTION and MANAGEMENT: Air Pollution: Criteria pollutants – Ozone, Particulate Matter, Carbon Monoxide, Nitrogen, Oxides, Sulphur Dioxide, Lead; Acid Rain Cycle. Greenhouse effect. Global warming. Water as a resource; Lentic and Lotic Water Systems; Rain Water Harvesting; Water pollution: Potable and waste water Pollution and management; Noise pollution-sources and effects of noise; Municipal Solid Waste: Hazardous Waste: Electronic Waste: Biomedical Waste; Solid Waste Management: Landfills, composting Process.

For CIA only

UNIT – III	**06 Hours**

ENERGY: Energy Types of energy: Conventional sources of energy, fossil fuel, Coal, Solar, wind; Non-conventional Sources of Energy, Biofuels - biomass, biogas.

For CIA	
UNIT – IV	**06 Hours**
DISASTERS & MANAGEMENT: Definition, Natural (Earthquakes, landslides, floods), Man-made disasters (biological, chemical, nuclear, radiological explosions) – definition, causes and management and/or mitigation strategies; Bhopal & Chernobyl Disasters. **For CIA only**	
UNIT – V	**06 Hours**
ENVIRONMENTAL IMPACT ASSESMENT: Environmental Impact Assessment (EIA); Air pollution monitoring and Ambient Air Quality Standards (AAQS); Environment Protection Act, 1986. Water (Prevention and control of Pollution) Act .Wildlife Protection Act .Forest Conservation Act .Issues involved in enforcement of environmental legislation. Public awareness. **For CIA only**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Critically elucidate the basic concepts that govern environmental quality, ambient air quality standards	L3
2	Compare different Energy resource and their environmental implications	L3
3	Identify different types of pollution, waste streams	L3
4	Identify different natural and manmade disasters and prevention	L4
5	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	3
CO1	2	2	-	1	-	3	2	-	-	-	1	1	-	3
CO2	2	2	1	1	-	3	2	-	-	-	1	1	-	3
CO3	2	3	1	1	-	3	2	-	-	-	1	1	-	3
CO4	2	2	2	2	-	3	2	1	-	-	1	1	-	3
CO5	2	3	2	2	1	3	2	1	-	-	2	1	-	3

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

TEXT BOOKS:

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.

REFERENCE BOOKS:

1. P. Aarne Vesilind, Susan M.Morgan, Thomson (2008). "Introduction to Environmental Engineering" (2008), Thomson learning, Second Edition, Boston.
2. R. Rajagopalan (2005). "Environmental Studies – From Crisis to Cure" Oxford University Press, New Delhi.

****

KANNADA KALI

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

SEMESTER – I/II

Subject Code	: 25ENG1112	**Credits**	: 01
Hours / Week	: 01 Hours	**Total Hours**	: 15 Hours
L–T–P–S	: 1–0–0–0		

Course Learning Objectives:

This cource enables students:

1. To introduce Kannada language & culture to Non-Kannada speakers.
2. To train them to communicate in colloquial kannada with connivance.

Teaching-Learning Process (General Instructions):

These are some of the innovative pedagogical approaches to accelerate the attainment of the various course outcomes.

1. **Lecture method**: means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching: Active learning** that includes brainstorming, group work, formulating questions, notetaking, and annotating.
3. Show **Video**s to explain and illustrate the various concepts.
4. Encourage **Collaborative** learning in the class.

Unit I	07 Hours
Introduction to Karnataka & kannada culture, evolution of Kannada Introduction to Kannada Alphabets, Introduction to Kannada Numbers	
Unit II	08 Hours
Kannada words, Sentences & Phrase making for colloquial communication, Tenses, Pronoun, verbs,	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Developing the ability to understand and speak Kannada, read and write the Kannada script, and communicate effectively in day-to-day situations.	L1
2	Gaining confidence in using the language and interacting with Kannada speakers.	L2
3	Expanding vocabulary and understanding basic Kannada grammar rules.	L3
4	Improving listening comprehension and developing the ability to engage in simple conversations.	L4
5	Learning to read and write the Kannada alphabet and basic words and sentences.	L3

Text books:

1. Kannada Kali – Dr. Lingadevaru Halemane.
2. Kannada Paatagalu – Editor : Dr. Chandrashekara kambara
3. SLN Sharma & K Shankaranarayana "Basic Gramer", Navakarnataka Publication.
4. Spoken Kannada. Publication : Kannada Sahitya Parishat Bangalore.

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**	**3**
CO1	-	-	-	-	-	1	2	-	3	-	2	-	-	1
CO2	-	-	-	-	-	1	2	-	3	-	2	-	-	1
CO3	-	-	-	-	-	1	2	1	3	-	2	-	-	1
CO4	-	-	-	-	-	2	2	1	3	-	2	-	-	2
CO5	-	-	-	-	-	1	2	1	3	-	3	-	-	2

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

Activity Based Learning (Suggested Activities in Class)

1. Using worksheets for tasks like circling the odd one out, drawing objects, or coloring pictures.
2. Encouraging group work through activities like rotating through different learning boards with questions, where each group writes their answers and then rotates to the next board.
3. Activities involving learning and writing Kannada words.

****

C PROGRAMMING FOR PROBLEM SOLVING

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

SEMESTER – I/II

Course Code: 25ENG1114		**Credits**	: 04
Hours/Week: 05 Hours		**Total Hours**	: 60 Hours
L–T–P–J	: 2–1–2–0		

Course Learning Objectives:

This Course will enable students to:

1. **Elucidate** the basic architecture and functionalities of C programming language.
2. **Apply** programming constructs of C language to solve the complex problems
3. **Explore** data structures like arrays, structures, unions and pointers in implementing solutions to real world problems
4. **Design** and Develop Solutions to problems using structured programming constructs such as functions.

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

UNIT – I	**06 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.

(Text Book-1: Chapter 2 & Chapter 3)

UNIT – II	**06 Hours**

Arrays: Introduction, declaration & initialization of array, reading and writing array elements, Operations on array: Traversal, searching (Linear and Binary search), sorting (Bubble sort and Selection Sort). Declaration and Initialization of two-dimensional arrays. Matrix Operations (addition, subtraction, multiplication, transpose) using two-dimensional array.

Strings: Definition, declaration, initialization, and representation. String handling functions and character handling functions.

(Text Book-1: Chapter 5:5.1 to 5.9 & Chapter 6)

UNIT – III	06 Hours
Pointers: Definition and declaration and initialization of pointers. Accessing values using pointers. Accessing array elements using pointers. **Functions:** Definition and declaration. Built-in functions and User-defined functions. Categories of functions with example. Pointers as function arguments, array as function argument, Call-by-value and call-by-reference. Recursion. ***(Text Book-1: Chapter 7: 7.1 to 7.17 & Chapter 4:4.1 to 4.8, 4.10)***	
UNIT – IV	**06 Hours**
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. ***(Text Book-1: Chapter 8: 8.1, 8.2,8.6)***	
UNIT – V	**06 Hours**
Memory allocation in C programs: Dynamic memory allocation, memory allocation process, allocating a block of memory, releasing the used space, altering the size of allocated memory. **Files**: Defining, open, read, write, seek and closing of both textual and random files. ***(Text Book-1: Chapter 7: 7.18 to 7.20 & Chapter 9: 9.1 to 9.5, 9.8)***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** fundamental programming constructs such as data types, operators, control flow, and loops to develop basic C programs.	L2, L3
2	**Demonstrate** the ability to use one-dimensional and two-dimensional arrays, perform matrix operations, and manipulate strings using standard C functions.	L3
3	**Apply** pointer concepts and function mechanisms including recursion, call-by-value/reference, and arrays as arguments to build modular C programs.	L3
4	**Develop** programs using structures and unions to represent and manage compound data types for real-world applications like employee or student databases.	L3
5	**Implement** dynamic memory allocation techniques and file operations in C for efficient data storage and retrieval.	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	3
CO1	3	2	2	1	2	-	-	-	-	-	1	2	1	-
CO2	3	3	2	2	3	-	-	-	-	-	1	2	2	-
CO3	3	3	3	2	3	-	-	-	-	-	1	2	2	-
CO4	3	3	3	3	3	-	-	1	-	-	2	2	3	-
CO5	3	3	2	3	3	-	1	1	-	-	2	2	3	1

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

TEXT BOOKS:

1. Reema Thareja, "Programming in C". Oxford University Press, Second Edition, 2016.

REFERENCE BOOKS:

1. Brian W. Kernigham and Dennis M. Ritchie, (2012) "The C Programming Language", 2nd Edition, PHI.
2. Behrouz A. Forouzan, Richard F. Gilberg, "Computer Science - A Structured Approach Using C", Cengage Learning, 2007.
3. Vikas Gupta, "Computer Concepts and C Programming", Dreamtech Press 2013.

E-Resources:

1. https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem-solving methods.
2. https://www.w3schools.com/c/index.php
3. https://www.guvi.in/courses/web-development/c-programming/
4. https://www.tutorialspoint.com/cprogramming/index.htm
5. https://pythontutor.com/

Activity Based Learning (Suggested Activities in Class)

1. Demonstration of solution to a problem through designing the Flowchart or any design notations using **draw.io** in the group of four and justify using snippets or algorithms.

C PROGRAMMING FOR PROBLEM SOLVING LABORATORY (Total Hours: 26)

1. Programming Basics: Swapping Numbers, Simple Interest, and Factorial.
2. Quadratic Equation Solver
3. Number Operations: Palindrome Check and Power Calculation.
4. Fibonacci Series and Greatest Common Divisor (GCD) Calculation.
5. Calculator Emulation
6. String Manipulation
7. Sorting an Array of Integer Elements.
8. Searching an Array of Elements.
9. Pointer Demonstration using Functions.
10. Case Study on Strings and Functions.

****

PHYSICS FOR COMPUTER SCIENCE CLUSTER

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

SEMESTER – I/II

Course Code	:	25EN1115	**Credits**	:	04
Hours / Week	:	05 Hours	**Total Hours**	:	60 Hours
L–T–P–J	:	3–0–2–0			

Course Learning Objectives:

This Course will enable students to:

1. Understand the fundamental principles of quantum mechanics and apply them to analyze and solve engineering problems in science and technology.
2. Explore the foundational concepts of quantum information and their applications in the emerging field of quantum computing.
3. Learn the principles of quantum optics, including the operation and applications of lasers and optical fiber communication systems in engineering.
4. Comprehend the band structure of solids and analyze the electrical properties and device applications of semiconductors.
5. Understand thin-film phenomena, examine various thin-film fabrication techniques, and distinguish between top-down and bottom-up approaches for synthesizing nanomaterials, with emphasis on their applications in engineering and medicine

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** means it includes not only traditional lecture method, but different type 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 rather than simply recall it.
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**

Introduction to Quantum mechanics:

Failure of Classical mechanics, Planck's hypothesis - Principles of quantum Mechanics (Wave-Particle Duality and Uncertainty Principle), Wave function, Eigen function and Eigen values, Schrodinger time independent wave Equation in 1D (Derivation), Application - Particle in a potential well (Derivation), Numerical.

UNIT – II	**09 Hours**

Quantum Information and Computing

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 and Fiber Optic Communication

- **Quantum Optics:** Process of absorption, spontaneous and stimulated emission Einstein coefficients (qualitative), Pumping, population inversion, resonators, Laser characteristics, beam quality, characteristics of laser, Types of lasers: construction and working of solid-state (Nd:YAG) and semiconductor laser - Applications of Lasers in engineering. **(5 hours)**
- **Fiber Communication:** Basics of optical fibers and light propagation, Numerical aperture, acceptance angle, Step-index and graded-index fibers, Fiber losses: attenuation, dispersion, bending, Applications in communication and sensing. **(4 hours)**

UNIT – IV	09 Hours

- **Semiconductor Physics:** Classification of Solids, Band structure, Fermi level in intrinsic and extrinsic semiconductors, Fermi-Dirac distribution function, Expression for concentration of electrons in conduction band (Derivation), Intrinsic carrier concentration, Conductivity of semiconductors, Hall effect, Derivation of Hall Coefficient, introduction to quantum hall effect - Numerical. **(6 hours)**
- **Semiconductor Devices:** Zener diode: breakdown mechanisms and applications, Light Emitting Diode (LED): working principle and characteristics. Photodiode, Solar Cell and BJT: basic operation and I-V characteristics. **(4 hours)**

UNIT – V	09 Hours

Thin films and Nano technology

- **Thin Films Technology:** Introduction to thin films-Advantages of thin films over bulk materials. Introduction to thin film deposition processes - Physical vapor deposition (Thermal evaporation technique and sputtering technique) process, Applications of Thin films**. [3 hours]**
- **Nano Science & Technology:** Introduction to Nano materials, Classification of nano materials, Size dependent properties of materials (Quantum confinement), Top-down and Bottom-up approach- Ball-milling, and Photolithography Process. Applications of Nano technology in medicine and Engineering. **[3 hours]**

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Explain** the fundamental postulates of quantum mechanics and apply them to simple physical systems relevant to engineering applications.	L2, L3
2	**Analyse** basic concepts of quantum information, quantum bits (qubits), and evaluate simple quantum processing and their computational advantages.	L3
3	**Illustrate** the working principles of lasers and optical fibers, and apply them to modern communication systems and other engineering fields.	L3
4	**Interpret** the band theory of solids and characterize the electrical behavior of semiconductors and their applications in devices.	L3
5	**Compare** various thin-film fabrication techniques, explain nanoscale material synthesis approaches, and evaluate their applications in engineering and medical fields.	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	3
CO1	3	2	1	1	2	-	-	-	-	-	1	1	1	-
CO2	3	3	2	2	2	-	-	-	-	-	1	2	1	-
CO3	3	3	3	2	3	-	-	-	-	-	1	2	2	-
CO4	3	3	2	3	3	1	-	-	-	-	2	2	2	1
CO5	3	2	3	3	3	1	1	1	-	-	2	2	3	2

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

TEXT BOOKS:

1. R. Shankar, "Principles of Quantum Mechanics", Springer, 2nd Edition, 1994.
2. Michael A. Nielsen & Isaac L. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 10th Anniversary Edition, 2010.
3. Mark Fox, "Quantum Optics", Oxford University Press, Oxford Master Series in Physics, 3rd Edition, 2022.
4. K. Thyagarajan & A.K. Ghatak, "Lasers: Fundamentals and Applications", Springer India, 2nd Edition: 2010.

REFERENCE BOOKS:

1. Gerd Keiser, "Optical Fiber Communications", Springer, 5th Edition, 2021.
2. Ajoy Ghatak & K. Thyagarajan, "Introduction to Fiber Optics", Cambridge University Press, 1st Edition, 1998.
3. S.O. Kasap, "Principles of Electronic Materials and Devices", McGraw-Hill Education, 4th Edition, 2018.
4. Arthur Beiser, "Concepts of Modern Physics", McGraw-Hill Education, 6th Edition, 2003.

E-Resources:

1. NPTEL lectures on Quantum Computing & Semiconductor Physics and Devices

Activity-Based Learning (Suggested Activities in Class)

1. Use physical models and simulations to visualize key physics principles behind real-world engineering systems, like oscillations, interference, diffraction, etc.
2. Role-play or storytelling sessions where students simulate quantum computing logic gates or quantum key distribution protocols.
3. Hands-on circuit-building exercises using breadboards to understand working of devices like LEDs, photodiodes, and BJTs.
4. Demonstration using optical fibers to show light propagation, total internal reflection, and loss mechanisms.
5. Interactive videos and animations to demonstrate thin film deposition methods like CVD, PVD, and sol-gel techniques.
6. Regular use of virtual lab platforms (IIT/NPTEL/PhET) to simulate experiments that are otherwise difficult in a physical lab setting.

PHYSICS- COMPUTER SCIENCES LABORATORY (Total Hours: 26)

1. **I-V Characteristics of a Zener Diode:** I-V characteristics of a Zener Diode in forward and reverse bias conditions
2. **Planck's Constant:** Determination of Planck's Constant using LED
3. **Band Gap Energy:** Determination of the energy gap of an intrinsic semiconductor
4. **Transistor Characteristics:** Input and output characteristics of NPN Transistor in C-E configuration
5. **Diffraction Grating:** Determination of wavelength of a laser light using a diffraction grating
6. **Fiber optic system:** Determination of numerical aperture, acceptance angle, and attenuation of an optical fiber

7. **Four Probe Method:** Determination of resistivity of a material using the Four-probe technique
8. **Hall Effect:** Determination of carrier concentration and mobility of a semiconductor
9. Realization of the Hadamard Gate and its Bloch sphere visualization using Qiskit/QuUniverse Quantum computing programming.
10. Realization of the C-NOT Gate and its Bloch sphere visualization using Qiskit/Quniverse Quantum computing programming.

****

ENGINEERING SCIENCE COURSE – II FOR COMPUTER SCIENCE CLUSTER
INTRODUCTION TO ELECTRICAL AND ELECTRONICS ENGINEERING

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

SEMESTER – I/II

Subject Code	:	25ENG1109	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P-J	:	3–0–0-0			

Course Learning Objectives:

1. To introduce fundamental electrical quantities such as charge, current, voltage, power, and energy, and their relationships in basic circuits.
2. To enable students to apply Ohm's Law and Kirchhoff's Laws to analyze simple resistive, series, and parallel electrical networks.
3. To impart a foundational understanding of AC and DC waveforms, including key parameters like RMS, peak, and average values.
4. To develop conceptual understanding of magnetic circuits and electrical machines, including transformers and DC machines, along with their working principles.
5. To familiarize students with the characteristics and applications of diodes and transistors, particularly in rectifier and switching circuits.
6. To introduce operational amplifiers and their basic configurations such as inverting, non-inverting, and summing amplifiers.
7. To provide a conceptual understanding of digital logic circuits, including logic gates, flip-flops, and binary counters.
8. To provide a high-level overview of microcontroller systems, focusing on block diagram architecture and their role in embedded 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** means it includes not only traditional lecture method, but different type 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 analyse information rather than simply recall it.
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.

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.

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.	
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.	
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.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Demonstrate a clear understanding of the basic principles of electrical and electronic components**, including resistors, diodes, transistors, operational amplifiers, logic gates, and microcontrollers.	L2
2	**Apply basic circuit law's and priciples to analyze electrical circuits**, including resistive networks, AC/DC waveforms, and basic transformer and DC machines	L3
3	**Apply knowledge of semiconductor devices**, such as diodes and transistors, to explain rectification and switching behavior in electronic circuits.	L3
4	**Apply the principles of operational amplifiers** to configure basic circuits such as inverting, non-inverting, and summing amplifiers.	L3
5	**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	3
CO1	3	2	1	1	2	1	-	-	-	-	1	1	1	-
CO2	3	3	2	2	3	1	-	-	-	-	1	1	2	-
CO3	3	3	3	2	3	1	-	1	-	-	2	1	2	-
CO4	3	3	3	3	3	1	1	1	-	-	2	2	3	1
CO5	3	2	3	3	3	2	1	1	1	1	2	2	3	2

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.

E-Resources:

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/

Activity-Based Learning (Suggested Activities in Class):

1. Simulation of different electronic circuits. E.g., Rectifiers and Amplifiers using LtSpice.

****

INTRODUCTION TO SUSTAINABLE ENGINEERING

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

Semester I/II

Course Code	:	25ENG1120	**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30
L–T–P–S	:	2–0–0–0			

Course Learning Objectives:

This course will enable students to:

1. Introduce the foundational concepts of sustainable engineering and its role in addressing environmental and resource challenges.
2. Develop the ability to analyze energy and material flows in engineering systems and propose efficient conservation strategies.
3. Apply green design and sustainable manufacturing principles for minimizing environmental impacts across product life cycles.
4. Integrate sustainable approaches in mobility, infrastructure, and construction while balancing performance, safety, and environmental considerations.
5. Utilize digital tools, lifecycle assessment, and responsible innovation frameworks in making ethical and sustainable engineering decisions.

Teaching-Learning Process (General Instructions):

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

1. ***Lecture method*** means it includes not only traditional lecture method but different *types of teaching methods* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the 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, and develops thinking skills such as evaluating, generalizing, and analysing information rather than simply recalling it.
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 student's understanding.
9. **Practical experimentation** of different additive manufacturing methods

UNIT – I	**06 Hours**

Foundations of Sustainable Engineering

Sustainability principles in engineering systems, environmental and resource challenges. Systems thinking in design and analysis, lifecycle perspective, and circular economy concepts. Product stewardship and sustainable development goals as guiding frameworks. Understanding the role of engineers in sustainable transitions across sectors.

UNIT – II	**06 Hours**

Energy and Resource Efficiency: Energy flow in thermal, fluid, and structural systems, Energy conservation strategies and efficient design practices in physical systems. Material efficiency, sustainable sourcing, and reduction of embodied energy. Case studies on optimizing system performance for minimal resource use.

UNIT – III	06 Hours
Green Design and Manufacturing Design for environment (DfE), reuse, and ease of disassembly in product lifecycle. Sustainable manufacturing through low-emission machining and fabrication methods. Additive and subtractive processes with minimal material waste. Selection of processes and materials based on environmental performance.	
UNIT – IV	**06 Hours**
Sustainable Mobility and Infrastructure Light weighting and material substitution in transportation systems design. Structural optimization for energy savings and enhanced durability. Sustainable material choices in public infrastructure and construction. Balancing safety, performance, and environmental considerations in design. Net carbon zero in the built environment.	
UNIT – V	**06 Hours**
Digital Tools and Responsible Innovation Lifecycle Assessment (LCA) using digital tools for evaluating sustainability. Design optimization and predictive analytics for efficient system operation.AI-assisted support in material selection and maintenance planning. Engineering ethics, standards, and compliance with EPR, RoHS, and SDGs.	
DEMONSTRATION EXERCISE/ASSIGNMENTS	
1. **Demonstration of the Principle and Operation of a Turbo Machines** 2. **Demonstration of the Principle and Operation of a CNC Machine** – Discuss role in precision and low-waste manufacturing 3. **Demonstration of the 3D Printing Process** – Understand sustainable prototyping and material recyclability 4. **Demonstration of a Heat Transfer Application Device** (e.g., heat exchanger) and preparation of a report analyzing energy losses 5. **Demonstration of an Air Conditioning System** – Measure and analyze Coefficient of Performance (COP) and environmental impact 6. **Demonstration of Mechatronic System Elements** – Explore automation and energy-saving benefits 7. **Demonstration of a Robotic System** – Understand its role in smart and sustainable manufacturing 8. **Case study** on Net Zero Energy Building (NEZB) at Dayananda Sagar University	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Analyze engineering systems from a sustainability perspective and evaluate lifecycle and circular economy principles for improved system design.	L2
2	Assess energy flows and material usage in complex engineering systems to propose efficient conservation and resource optimization solutions.	L3
3	Design products and manufacturing processes that minimize emissions, waste, and environmental impact throughout the product lifecycle.	L3
4	Evaluate sustainable mobility and infrastructure solutions by integrating material selection, structural optimization, and durability	L3

	considerations.	
5	Apply digital tools, lifecycle assessments, and AI-supported methods to optimize system performance while ensuring compliance with ethical standards and sustainability regulations.	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	3
CO1	2	2	1	1	-	3	2	-	-	-	1	1	-	3
CO2	2	3	2	2	1	3	2	-	-	-	1	1	-	3
CO3	2	3	3	2	2	3	2	1	-	-	1	1	1	3
CO4	2	2	3	3	2	3	2	1	1	1	2	1	1	3
CO5	2	3	3	3	2	3	3	1	1	1	3	1	2	3

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

TEXT BOOKS :

1. David T. Allen, David R. Shonnard, *Sustainable Engineering: Concepts, Design and Case Studies*, 2nd Edition, Pearson Education, 2012.
2. Bhavik R. Bakshi, *Sustainable Engineering: Principles and Practice*, Cambridge University Press, 2019.

REFERENCES :

1. David Dornfeld (Ed.), *Green Manufacturing: Fundamentals and Applications*, Springer, 2013.
2. Michael F. Ashby, *Materials and the Environment: Eco-Informed Material Choice*, 2nd Edition, Elsevier, 2012.

****

BIOLOGY FOR ENGINEERS

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

SEMESTER – I/II

Subject Code	: 25ENG1121	**Credits**	: 02
Hours / Week	: 02 Hours	**Total Hours**	: 30 Hours
L–T–P–S	: 2–0–0–0		

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. Understand basic engineering principles imminently run physiological processes particularly about engineering designs and solutions are arrived citing body functional examples.
3. Explain aspects that many bio-solutions could be foundational to design, develop better processes, products and useful to achieve quality of life.

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*** means it includes not only traditional lecture method, but different *type of teaching methods* like power point presentations and group discussion 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. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT 1	**6 Hrs**
Biomimetics: Biology for Engineers, Body Fluid: Blood- Mechanics of heart, Blood pressure, Life molecules: Water, Carbohydrates, Proteins, Lipids and Nucleic acids, Biomimetics: Bio-processes - engineering analogies **For CIA only**	
UNIT 2	**6 Hrs**
Bioenergy: Unit of life: Human and Plant cell, Metabolism: Enzymes as Bio-catalysts and physiological entities, Anabolism- Bioenergy from Sun-Photosynthesis, catabolism **For CIA only**	
UNIT 3	**6 Hrs**
Biomechanics -Human Body Movement Mechanics: Normal Human Movement: Force-Vector of Body; Movement Angles; Muscle contraction -Relaxation; Posture – Static & Dynamic; Ideal and abnormal posture, Practical: Stepping-Lifting-Sit-Stand. **For CIA only**	
UNIT 4	**6 Hrs**
Bioelectronics: Brain & Computer: Senso-neural networks, Biosensors and IoT as applied to biology, Bionic Eye: Mechanism of Vision, Electronic Nose: Bio-olfactory mechanisms (Science of smell), Impulses: Cardiac and Nerve, Biological Clock and Circadian rhythm **For CIA only**	
UNIT 5	**6 Hrs**
Biopharma: Metabolic syndromes, Cancer and its diagnostics, Lab on a chip, Drug development and Discovery **For CIA only**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**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
2	**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
3	**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

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	3
CO1	3	2	1	1	1	1	1	-	-	-	1	2	1	2
CO2	3	3	2	2	1	1	1	-	-	-	1	2	1	2
CO3	3	3	3	2	2	1	1	-	-	-	1	3	2	2

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

TEXT BOOKs:

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

REFERENCE BOOKS:

1. 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
2. David Nelson, Michael Cox. "Lehninger Principles of Biochemistry". W H Freeman & Company, Seventh Edition, 2017.
3. Janine M Benvus. "Biomimicry: Innovation inspired by Nature". William Morrow Paperbacks, 2002.
4. Lecture Notes, PPT slides by course instructor.

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects, taking students on an educational tour.

****

CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS

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

SEMESTER – I/II

Course Code	: 25ENG1122	**Credits :**	01
Hours / Week	: 01 Hours	**Hours :**	15 Hours
L-T-P-J	: 1-0-0-0		

Course Learning Objectives:

This Course will enable students to **:**

1. Acquaint the students with legacies of constitutional development in India and help those to understand the most diversified legal document of India and philosophy behind it.
2. Make students aware of the theoretical and functional aspects of the Indian Parliamentary System.
3. Channelize students thinking towards basic understanding of the legal concepts and its implications for engineers.
4. Acquaint students with latest legislation and Laws with related regulatory framework.

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** means it includes not only traditional lecture method, but different type of teaching methods like power point presentations and group discussion 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, note taking, 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. Discuss how every **concept can be applied to the real world** - and when that's possible, it helps improve the students' understanding.

COURSE CONTENT

UNIT 1: Introduction and Basic Information about Indian Constitution	**3 Hrs**
Meaning of the constitution and constitutionalism; Historical background of the constituent assembly; Indian constitution and its salient features; The sources of Indian constitution; & The Preamble of constitution.	
UNIT 2: Fundamental rights & duties and D.P.S.P	**3 Hrs**
Importance and features of fundamental rights; Types of fundamental rights; Directive principles of state policy; Fundamental duties.	
UNIT 3: Judiciary	**3 Hrs**
The independence of the judiciary; Appointment of judges; Judicial review and judicial activism; & Public interest litigation.	
UNIT 4: Amendment and emergency	**3 Hrs**
Amendment of the constitutional power and procedure; The historical perspective of the constitutional amendments in India; Emergency provisions; National emergency; State emergency; & Financial	

emergency.	
UNIT 5: Intellectual property rights	**3 Hrs**
Overview of the Concept of Intellectual Property; Importance of Human Creativity in Presence Scenario; Patents; Trademark; & Copyrights.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Identify and explore the basic features and modalities about Indian constitution.	L2
2	Differentiate and relate the functioning of Indian parliamentary system at the Centre and State level.	L3
3	Differentiate different aspects of Indian Legal System and its related bodies.	L3
4	Discover and apply different laws and regulations related to engineering practices.	L3
5	Correlate role of engineers with different organizations and governance models	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	3
CO1	1	2	-	-	-	3	3	1	2	-	1	-	-	2
CO2	1	2	1	-	-	3	3	1	2	-	1	-	-	2
CO3	1	2	2	1	-	3	3	2	2	1	1	-	-	2
CO4	1	2	2	1	1	3	3	2	3	1	2	-	-	3
CO5	1	2	1	1	1	3	3	2	3	1	3	-	-	3

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

TEXT BOOKs:

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

REFERENCE BOOKS:

1. V.N.Shukla's Constitution of India by Mahendra Pal Singh, 13nth edition published by Eastern Book Company.
2. Intellectual property rights law in India 3rd edition by T.Ramappa published by Asia law house.
3. Managing Intellectual property by, 3rd edition Vinod.V.Sople published by PHI learning.

E-RESOURCES:

1.https://www.constitutionofindia.net/read/

2.https://www.constitutionofindia.net/debate_paragraphs/
3.https://dst.gov.in/sites/default/files/E-BOOK%20IPR.pdf
4. https://ipindia.gov.in/resources.htm

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

SINGLE AND MULTIVARIATE CALCULUS (SMC)

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

SEMESTER – II

Subject Code	: 25ENG1201	**Credits** :	04
Hours / Week	: 03 Hours	**Total Hours** :	60 Hours
L–T–P–J	: 3–1–0-0		

Course Learning Objectives:

This Course will enable students to:

1. **Apply** sophisticated techniques of differential calculus to solve problems involving functions of multiple variables.
2. **Apply** double and triple integrals in various coordinate systems (Cartesian, polar, cylindrical, and spherical) and effectively employ them to calculate areas, volumes.
3. **Acquire** a comprehensive understanding of fundamental concepts related to functions of multiple variables, including limits, continuity, and partial derivatives.
4. **Analyze** critical points of functions of two or more variables using partial derivatives and Lagrange multipliers, evaluate extreme values.
5. **Apply** vector calculus principles, such as line integrals, surface integrals, and the divergence theorem effectively to vector field.
6. **Analyze** the convergence and divergence of sequences and infinite series of real numbers by employing various convergence criteria and tests.

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** means it includes not only traditional lecture method, but different type 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 analyse information rather than simply recall it.
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	**12 Hours**

Differential Calculus

Functions of two or more variables: Definition, Region in a plane, Level curves, Level surfaces, Limits, Continuity, Partial derivatives, Differentiability, Extreme values and saddle points, Lagrange multipliers.

Self-Learning Component: Single variable calculus

UNIT – II	**12 Hours**

Integral calculus

Double integral and iterated integrals - Cartesian and polar coordinates, Triple integral, Change of variables, Multiple integrals in cylindrical and spherical coordinates.

UNIT – III	12 Hours
Vector Calculus Line Integrals, Vector Fields, Work, Circulation and flux, Path independence, Potential functions, and Conservative fields, Green's theorem in the plane, Surface area and surface integrals, Surface area of solid of revolution, Parametrized surfaces, Stokes' theorem, The Divergence theorem.	
UNIT – IV	**12 Hours**
Sequence and Series I: Sequences of real numbers and their convergence criteria, Infinite series, Sequence of partial sums, Tests for convergence/divergence - nth term test, Boundedness and monotonicity, Integral, Condensation, Comparison, Ratio and root tests.	
UNIT – V	**12 Hours**
Sequence And Series II: Alternating series, Absolute and conditional convergence, Rearrangement theorem, Power series, Taylor and Maclaurin series (one and two variables).	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply the principles of differential calculus to solve problems involving functions of two or more variables.	L2
2	Utilize double and triple integrals in Cartesian, polar, cylindrical, and spherical coordinates to compute areas, volumes, and evaluate mathematical expressions.	L3
3	Extend a comprehensive understanding of the concepts related to functions of multiple variables and evaluate critical points, including extreme values and saddle points using partial derivatives and Lagrange multipliers.	L3
4	Analyze vector calculus concepts, such as line integrals, surface integrals, and the divergence theorem, in the context of vector fields and their applications.	L3
5	Apply convergence criteria and various tests, such as the nth term test, boundedness and monotonicity, integral, condensation, comparison, ratio, and root tests, to analyze and determine the convergence or divergence of sequences and infinite series of real numbers.	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	3
CO1	3	3	1	1	1	-	-	-	-	-	1	1	-	-
CO2	3	3	2	2	1	-	-	-	-	-	1	1	-	-
CO3	3	3	2	2	2	-	-	-	-	-	1	2	1	-
CO4	3	3	3	2	2	-	-	-	-	-	2	2	1	-
CO5	3	3	3	3	2	-	-	-	-	-	2	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. Advanced Engineering Mathematics, Erwin Kreyszig , 2015, 10th Edition, Wiley India.

REFERENCE BOOKS:

1. Higher Engineering Mathematics, B.S. Grewal, 2015, 43rd Edition, Khanna Publishers.
2. Higher Engineering Mathematics, John Bird, 2017, 6 th Edition, Elsevier Limited.
3. Calculus: Early Transcendentals, James Stewart, 2017, 8 th edition, Cengage Learning.
4. Engineering Mathematics, K.A. Stroud and Dexter J. Booth, 2013, 7 th Edition, Palgrave Macmillan.
5. Basic Multi Variable Calculus, Marsden, Tromba and Weinstein, W.H. Freeman, 1993 Third Edition.

E-Resources:

1. https://www.youtube.com/playlist?list=PLtKWBwryn4nA2h8TFxzWL2zy8O9th_fy
2. https://www.youtube.com/playlist?list=PLU6SqdYcYsfJqbZvQECrwnlQrp4f_g6isX

Activity Based Learning (Suggested Activities in Class)

1. Activity which makes students to apply the concepts learned in the course to apply in real life engineering problems will be discussed in class.
2. Real world problem solving and puzzles using group discussion.
3. Demonstration of solution to a problem through programming.

****

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

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

Department of Computer Science & Medical Engineering

SCHOOL OF ENGINEERING

Department of

Computer Science & Medical Engineering

SYLLABUS 2nd YEAR

B.Tech. PROGRAMME– 2025-29 BATCH

Academic Year 2025-26

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

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

APPLIED STATISTICS & STOCHASTIC PROCESSES

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2301	**Course Type:** Basic science Course
Programs Offered: CSME	**Total Hours:** 45 Hours
Semester: III	**Hours/Week:** 03 Hours
Credits: 03	**L – T – P – J:** 3 – 0 –0 – 0
Prerequisites: Calculus, probability theory, random variables and probability distributions, statistics, and linear algebra	

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.

Course Description:

This course provides a comprehensive understanding of probability, statistics, and stochastic processes with emphasis on analytical techniques and practical applications in engineering and data-driven decision making. It introduces random variables, probability distributions, and statistical measures, followed by correlation and regression analysis for studying relationships between variables. The course also covers hypothesis testing, parametric and non-parametric inference methods, and analysis of variance to support decision making under uncertainty. Further, students are introduced to stochastic processes including Markov chains, Poisson processes, and Monte Carlo simulation. The course enables learners to apply probabilistic and statistical methods in areas such as reliability engineering, quality control, forecasting, financial modelling, and system performance analysis.

Course Outcomes (CO)

CO	Course Outcomes	Bloom taxonomy level
CO1	Apply probability concepts and random variables to analyze uncertainty	L3
CO2	Analyze data using correlation and regression techniques	L4
CO3	Perform hypothesis testing and statistical inference for decision making	L4
CO4	Apply non-parametric tests for categorical data analysis	L3
CO5	Model and analyze stochastic processes such as Markov chains	L5

Course Modules

Module 1: Random Variables and Probability Distributions [05 Hours]
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

Module 2: Bi-variate Data, Correlation and Regression Analysis [12 Hours]
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

Module 3: Inference for Decision Making – I [12 Hours]
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

Module 4: Inference for Decision Making – II [10 Hours]

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

Module 5: Stochastic Processes [06 Hours]
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

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.

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

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

Authentic Intelligence:

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

RECOMMENDED TOOLS

1. Python
2. R Programming
3. NumPy
4. Pandas
5. Matplotlib
6. 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	Compute Karl Pearson's and Spearman's	R Programming	1 h	CO2

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Random Variables and Probability Distributions (03 Hours)					
	Analysis	rank correlation			
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

SUGGESTED READING

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. 2019
10. Trivedi, K. S. (2016), Probability & Statistics with Reliability, Queuing & Computer Science Applications (2nd edition), New Jersey, John Wiley & Sons, Inc.

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	3
CO1	3	2	-	-	2	-	-	-	-	1	-	2	-	2
CO2	3	3	2	2	3	-	-	-	1	1	-	3	2	3
CO3	3	3	3	2	3	-	-	-	1	2	-	3	2	3
CO4	3	2	2	2	2	-	-	-	-	1	-	2	2	2
CO5	3	3	2	3	3	-	-	1	-	1	2	3	2	3

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

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

INTRODUCTION TO BIO MEDICAL SIGNAL PROCESSING

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

Course Code: 25SM2302	**Course Type:** Mandatory Core Course
Programs Offered: CSME	**Total Hours:** 45 Hours
Semester: III	**Hours/Week:** 03 Hours
Credits: 03	**L – T – P – J:** 3 – 0 –0 – 0
Prerequisites: Programming Fundamentals (C# with Unity, C++, basic scripting and app development)	

Course Vision

- To provide students with a comprehensive overview of the computational and mathematical methods used in medical image processing.
- To enable the study of current methods used to enhance and extract useful information from medical images.
- To promote the understanding of imaging principles and fundamental equations in the biomedical domain.

Course Objectives

Students will:

1. Grasp various noise models and restoration techniques to gain specialized knowledge.
2. Explain the fundamental equations and imaging principles of biomedical image processing.
3. Adopt Problem-Based Learning to foster analytical skills and the ability to evaluate and generalize information.
4. Develop Critical Thinking by addressing higher-order questions regarding medical image data.

Course Outcomes (CO)

CO	Course Outcomes	Bloom taxonomy level
CO1	Students will be able to identify major processes involved in the formation of medical images and recognize specific imaging modalities (X-ray, MRI, CT) from their visualization.	L3
CO2	Students will be able to appraise the efficacy and drawbacks of various image segmentation techniques applied to clinical data.	L4
CO3	Students will gain the ability to understand and distinguish between different imaging functions and their respective spectral contents.	L4
CO4	Students will be able to discuss different image degradation models and implement various filtering techniques (spatial and frequency domain) to enhance medical image quality.	L3
CO5	Apply dimensionality reduction and statistical techniques such as PCA, ICA, correlation, and regression for efficient biomedical signal analysis, noise reduction, and feature extraction in medical imaging systems.	L5

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

Department of Computer Science & Medical Engineering

Broad Course Description

This course provides a comprehensive understanding of **biomedical signal processing** with a focus on the acquisition, analysis, and interpretation of physiological signals for healthcare applications. It begins with the fundamentals of bio-signal generation, types, and diagnostic parameters, followed by the study of electrode systems and signal conditioning techniques for accurate data acquisition.

The course further explores advanced **signal processing methods**, including Fourier and wavelet transform techniques, for analyzing both stationary and non-stationary biomedical signals. It introduces modern **machine learning approaches**, particularly Recurrent Neural Networks (RNNs) and their variants such as LSTM and GRU, for modeling and analyzing sequential biomedical data.

Additionally, the course covers **dimensionality reduction and statistical techniques** such as PCA and ICA for efficient feature extraction, noise reduction, and data representation. Emphasis is placed on real-world engineering applications, including medical imaging, diagnostic systems, and healthcare data analysis, enabling students to apply theoretical concepts to practical biomedical problems.

Course Modules

Module 1: Introduction to Biomedical Signal Processing **(09 Hours)**

Acquisition, Generation of Bio-signals, Origin of bio-signals, Types of bio-signals, Study of diagnostically significant bio-signal parameters.

Text book1: Chapter 1.1 to 1.3, Pages 1-69

Engineering Applications: Medical imaging system design, diagnostic imaging, microscopy-based analysis, breast cancer screening, biomedical instrumentation, and image-based clinical assessment.

Module 2: Electrodes for bio-physiological sensing and conditioning **(09 Hours)**

Electrode-electrolyte interface, polarization, electrode skin interface and motion artefact, biomaterial used for electrode, Types of electrodes (body surface, internal, array of electrodes, microelectrodes), Practical aspects of using electrodes, Acquisition of bio-signals (signal conditioning) and Signal conversion (ADC's DAC's) Processing.

Text book1: Chapter 3.4 to 3.4.5 & Text book2:

Engineering Applications: CT and MRI visualization, ultrasound imaging systems, nuclear imaging analysis, diagnostic decision support, computer-aided diagnosis, and healthcare imaging informatics.

Module 3: Transform Techniques **(09 Hours)**

Transform Techniques: Biomedical signal processing by Fourier analysis, Biomedical signal processing by wavelet (time frequency) analysis, Analysis (Computation of signal parameters that are diagnostically significant), Classification of signals and noise, Spectral analysis of deterministic, stationary random signals and non-stationary signals.

Text book1: Chapter 6.3 to 6.4 & 5.4 to 5.6 pages 349-430

Engineering Applications: Image acquisition optimization, digital image representation, medical image enhancement, contrast improvement, image quality assessment, and quantitative biomedical image analysis.

Module 4: Recurrent Neural Networks (RNNs) and Variants **(09 Hours)**

Sequential data and RNNs, limitations of vanilla RNNs, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), applications: text generation, sentiment analysis, time series.

Text book1: Chapter 9.9 to 10.2

Engineering Applications: Frequency-domain image analysis, image sharpness evaluation, resolution improvement, signal quality analysis, filter design, and performance measurement in biomedical imaging systems.

Module 5: Dimensionality reduction techniques **(09 Hours)**

Principal component analysis, Correlation and regression, Analysis of chaotic signals Application areas of Bio–Signals analysis Multi resolution analysis (MRA) and wavelets, Principal component analysis (PCA), Independent component analysis (ICA).

Text book1: Chapter 2, 5, 9

Engineering Applications: Medical image restoration, noise reduction in diagnostic images, reconstruction of degraded biomedical images, enhancement of CT/MRI/ultrasound images, filtering techniques in healthcare systems, and computer-aided medical image processing.

CASE STUDIES

CASE STUDY 1: NVIDIA Course Certification: Applications of AI for Anomaly Detection. **(Self-Paced Learning)**

Prepare data and build, train, and evaluate models using XGBoost, auto encoders, and GANs, Detect anomalies in datasets with both labelled and unlabeled data, Classify anomalies into multiple categories regardless of whether the original data was labelled.

Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-TC-01+V1

CASE STUDY 2: Students present a short research review.

Students review on at least one recent research topic such as:

- AI-based breast cancer detection using mammography
- Deep learning in MRI, CT, and ultrasound image analysis
- Biomedical image enhancement, restoration, and reconstruction techniques
- Computer-aided diagnosis using medical image processing
- Fourier transform and frequency-domain methods in biomedical imaging
- Noise reduction and filtering methods in degraded medical images
- Microscopy image analysis using machine learning approaches

Students present a short research review.

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration:**
  - **Module-1:** AI can enhance the study of biomedical images by enabling intelligent image interpretation, pattern recognition, and automated identification of features in body temperature imaging, microscopy, and X-ray-based breast cancer analysis.
  - **Module-2:** AI supports advanced biomedical imaging modalities such as tomography, nuclear medicine imaging, ultrasonography, and MRI through automated image segmentation, disease classification, computer-aided diagnosis, and improved medical decision support.
  - **Module-3:** AI improves image quality analysis by assisting in noise detection, contrast enhancement, histogram analysis, entropy-based feature extraction, and optimization of digitized biomedical image representation.
  - **Module-4:** AI strengthens Fourier-based biomedical image processing by enabling intelligent spectral analysis, blur estimation, signal-to-noise evaluation, and automated assessment of image sharpness and resolution.
  - **Module-5:** AI supports biomedical image restoration and reconstruction through predictive noise modelling, adaptive filtering, degradation estimation, and intelligent restoration techniques for improving the quality of diagnostic images.
- **Authentic Intelligence Reflection:**

- **Module-1:** This module reflects authentic intelligence by helping students understand the nature of biomedical images and their practical importance in identifying and analyzing real medical conditions such as breast cancer.
- **Module-2 :** This module reflects authentic intelligence by emphasizing the meaningful use of advanced imaging modalities and computer-aided diagnosis to support accurate, safe, and effective healthcare decisions.
- **Module-3:** This module reflects authentic intelligence by enabling students to evaluate image quality and information content in ways that directly improve the reliability and usefulness of biomedical image analysis.
- **Module-4:** This module reflects authentic intelligence by encouraging students to apply mathematical and signal-processing concepts for accurate interpretation of biomedical images and better diagnostic clarity.
- **Module-5:** This module reflects authentic intelligence by focusing on the responsible use of restoration and reconstruction techniques to enhance degraded medical images for real-world clinical applications.

Assessment Scheme

Professional Core Courses (Theory)

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

Recommended Tools

- **Programming:** Python / C# / C++/JavaScript/MATLAB
- **Visualization:** Blender / Maya/3ds Max/ParaView/Tableau/Power BI
- **AI Tools:** TensorFlow, PyTorch, Scikit-learn, OpenCV, MONAI, MediaPipe
- **Biomedical Imaging Tools:** MATLAB Image Processing Toolbok, ImageJ,3D Slicer, ITK, SimpleITK

Primary Textbooks

1. R. M. Rangayyan Biomedical Signal analysis: A Case study approach, IEEE press, JohnWiley & Sons. Inc, 2002.
2. Digital Image Processing, 4th Edition – Rafael C. Gonzalez, Richard E. Woods, Pearson, 2018

Additional Resources

1. Digital Image Processing for Medical Applications – Geoff Dougherty, Cambridge University Press, 2009
2. IEEE Journal of Biomedical and Health Informatics
3. MATLAB Onramp | Self-Paced Online Courses – MATLAB & Simulink
4. ImageProcessingPlace.com

AI-Powered Assignments

Assignment 1: AI-Generated Truth Table and Logic Circuit Evaluation

- **Students prompt an AI tool:** *"Generate truth tables and logic expressions for a given Boolean function: F(A,B,C) = A'B + BC' + ABC."*
- **Students must:**
    1. analyze the correctness of the generated truth table
    2. identify errors, missing min terms, or incorrect simplification

3. improve the solution using Karnaugh maps or Boolean algebra laws

- **Assessment focuses on human judgment, logical reasoning, and understanding of Boolean algebra.**

Assignment 2: AI-Generated Biomedical Image Classification Review

- Students prompt an AI tool:
  "Generate Python code for classifying biomedical images such as X-ray, MRI, ultrasound, or microscopy images."
  • Students must:
  1. analyze the correctness of the generated code
  2. identify errors, limitations, or inefficiencies
  3. improve the code for better classification accuracy and performance
- **Assessment focuses on human judgment, debugging ability, and understanding of biomedical image classification.**

Assignment 3: AI-Assisted Mammography Analysis Design

- Students prompt an AI tool:
  "Suggest a workflow for breast cancer detection using mammography images."
  • Students must:
  1. evaluate the medical relevance of the generated workflow
  2. identify missing diagnostic or preprocessing steps
  3. refine the workflow for better clinical usefulness
- **Assessment focuses on critical evaluation, medical relevance, and practical design improvement.**

Assignment 4: AI-Based Medical Imaging Modality Comparison

- **Students prompt an AI tool:**
  "Compare tomography, MRI, ultrasound, and nuclear medicine imaging for biomedical analysis."
  • Students must:
  1. verify the correctness of the comparison
  2. identify oversimplifications or inaccuracies
  3. Present an improved and application-oriented comparison
- **Assessment focuses on conceptual clarity, comparison skills, and domain understanding**.

Assignment 5: AI-Supported Image Quality Analysis

- Students prompt an AI tool:
  "Generate a method to analyze image quality parameters such as contrast, entropy, dynamic range, and histogram in biomedical images."
  Students must:
  1. assess whether the suggested method is technically correct
  2. identify gaps in quality evaluation
  3. Improve the approach for better biomedical image assessment
- Assessment focuses on analytical thinking and interpretation of image quality metrics

Assignment 6: AI-Generated Fourier analysis Workflow Review

- **Students prompt an AI tool:**
  "Generate a workflow for applying Fourier transform to biomedical images for frequency-domain analysis."
- **Students must:**
  1. analyze the feasibility and correctness of the workflow
  2. identify errors in spectral interpretation or implementation
  3. improve the workflow for meaningful biomedical image analysis
- **Assessment focuses on mathematical reasoning, signal analysis, and accuracy.**

Mini Project Component

- **Industry-Driven Projects**

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

Project 1 – AI-Based Breast Cancer Mammography Analysis

- **Problem:** Design a system that analyzes mammography images to identify suspicious regions and support early breast cancer screening.
- **Concepts used**: mammography analysis, image preprocessing, segmentation, feature extraction, AI-based classification
- **Industry relevance:** Radiology, cancer screening, computer-aided diagnosis

Project 2 – AI-Driven Biomedical Image Restoration and Reconstruction

- **Problem:** Develop a biomedical image restoration system that removes noise and blur from degraded medical images using intelligent filtering techniques.
- **Concepts used:** degradation models, noise modelling, inverse filtering, MMSE filtering, constrained least square filtering
- **Industry relevance:** Medical image enhancement, diagnostic support, radiology and healthcare systems

Project 3 – Wearable ECG Monitoring System with Motion Artifact Reduction

- **Problem:** Design a wearable system for continuous ECG acquisition that minimizes motion artifacts caused by electrode-skin interface changes during patient movement.
- **Concepts used:** Electrode-skin interface, motion artifact, polarization, biomaterials for electrodes, signal conditioning (amplifiers/filters), ADC/DAC conversion, stationary vs non-stationary noise analysis
- **Industry relevance:** Remote patient monitoring, ambulatory cardiac care, wearable health devices, telemedicine

Project 4 – Wavelet-Based Denoising of Non-Stationary EEG Signals

- **Problem:** Develop a signal processing system that removes physiological and environmental noise from EEG signals using time-frequency analysis for improved seizure detection.
- **Concepts used:** Non-stationary signal analysis, wavelet transform (time-frequency), classification of signals and noise, spectral analysis of random signals, extraction of diagnostically significant parameters
- **Industry relevance:** Neurology, epilepsy monitoring, brain-computer interfaces, intensive care units

Project 5 – LSTM-Based Prediction of Sepsis from ICU Time-Series Data

- **Problem:** Build a predictive system using sequential vital signs (heart rate, BP, temperature) to forecast sepsis onset hours before clinical diagnosis.
- **Concepts used:** Sequential data modeling, vanilla RNN limitations, LSTM networks, GRUs, time-series analysis, correlation and regression for feature selection
- **Industry relevance:** Intensive care units, early warning systems, emergency medicine, patient safety

Project 6 – PCA and ICA-Based Artifact Removal from Multi-Channel EEG

- **Problem:** Implement dimensionality reduction and blind source separation techniques to remove ocular and muscle artifacts from multi-channel EEG recordings.
- **Concepts used:** Principal Component Analysis (PCA), Independent Component Analysis (ICA), multiresolution analysis (MRA), analysis of chaotic biosignals, correlation and regression
- **Industry relevance:** Cognitive neuroscience, sleep studies, clinical neurophysiology, brain-computer interfaces

Project 7 – Fourier vs Wavelet Analysis for Heart Rate Variability (HRV) Assessment

- **Problem:** Compare spectral estimation methods (Fourier vs wavelet) to extract diagnostically significant HRV parameters for cardiac autonomic function assessment.
- **Concepts used:** Fourier analysis (deterministic/stationary signals), wavelet analysis (non-stationary signals), spectral analysis, classification of signals, diagnostically significant parameter computation

- **Industry relevance:** Cardiology, stress assessment, diabetic autonomic neuropathy screening, sports medicine

CO–PO–PSO Mapping Table

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**	**3**
CO1	3	2	-	-	2	-	-	-	1	-	-	2	1	1
CO2	3	3	2	1	3	-	-	-	1	1	-	3	2	2
CO3	3	3	3	2	3	-	-	-	1	1	-	3	3	2
CO4	3	3	3	3	3	1	-	-	2	1	1	3	3	3
CO5	3	3	2	3	3	1	-	-	2	2	1	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Week-by-Week Teaching Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to biomedical images	Lecture + examples	AI-assisted concept explanation	Understand the nature of biomedical images
	2	Body temperature as an image	Lecture + discussion	AI-based thermal image interpretation	Understand thermal imaging basics
	3	Transillumination in biomedical imaging	Lecture + examples	AI-generated imaging examples	Explain transillumination concepts
2	4	Light microscopy	Lecture + demo	AI-assisted cell/image recognition examples	Understand microscopy fundamentals
	5	Electron microscopy	Lecture + discussion	AI-supported image enhancement examples	Differentiate microscopy methods
	6	X-ray imaging, breast cancer, and mammography	Lecture + case examples	AI-based mammography analysis examples	Understand X-ray imaging applications
3	7	Tomography	Lecture + examples	AI-supported CT interpretation	Explain tomography principles
	8	Nuclear medicine imaging	Lecture + discussion	AI-assisted nuclear scan analysis	Understand nuclear imaging basics
	9	Ultrasonography	Lecture + demo	AI-generated ultrasound interpretation support	Explain ultrasound imaging concepts
4	10	Magnetic Resonance Imaging (MRI)	Lecture + examples	AI-assisted MRI analysis examples	Understand MRI fundamentals
	11	Objectives of biomedical	Lecture	AI-supported	Understand the purpose

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		image analysis		application mapping	of biomedical image analysis
	12	Computer-aided diagnosis (CAD)	Lecture + case study	AI-generated CAD workflow examples	Explain CAD concepts in healthcare
5	13	Difficulties in image acquisition and analysis	Lecture + discussion	AI-based artifact detection examples	Identify challenges in biomedical image acquisition
	14	Characterization of image quality	Lecture + examples	AI-assisted quality assessment	Understand image quality metrics
	15	Digitization of images: sampling and quantization	Lecture + problem solving	AI-generated digitization comparisons	Apply sampling and quantization concepts
6	16	Array and matrix representation of images	Lecture + coding demo	AI-assisted image matrix visualization	Represent biomedical images mathematically
	17	Optical density and dynamic range	Lecture + examples	AI-supported dynamic range analysis	Understand optical density and dynamic range
	18	Contrast, histogram, and entropy	Lecture + demo	AI-based histogram interpretation	Analyze image contrast and information content
7	19	Blur and spread functions	Lecture + examples	AI-assisted blur estimation	Understand blur characteristics in biomedical images
	20	Resolution in biomedical images	Lecture + discussion	AI-based resolution comparison	Explain image resolution concepts
	21	Fourier transform and spectral content	Lecture + demo	AI-supported spectral visualization	Understand frequency-domain analysis
8	22	Properties of Fourier transform	Lecture + examples	AI-generated transform demonstrations	Apply Fourier transform properties
	23	Modulation Transfer Function (MTF)	Lecture	AI-assisted MTF visualization	Understand image system performance
	24	Signal-to-noise ratio (SNR) and error-based measures	Lecture + problem solving	AI-supported noise analysis	Analyze signal and noise in biomedical images
9	25	Applications: image sharpness and acutance	Lecture + demo	AI-assisted sharpness evaluation	Evaluate image sharpness
	26	Model of image degradation/restoration process	Lecture + discussion	AI-based degradation prediction	Understand restoration framework
	27	Noise models in biomedical	Lecture +	AI-assisted noise	Identify different

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		images	examples	classification	biomedical noise models
10	28	Restoration in the presence of noise-only	Lecture + coding demo	AI-supported filtering suggestions	Understand basic restoration approaches
	29	Spatial filtering techniques	Lecture + demo	AI-assisted filter selection	Apply spatial filtering methods
	30	Periodic noise reduction using frequency domain filtering	Lecture + examples	AI-generated denoising workflows	Analyze frequency-domain restoration
11	31	Linear, position-invariant degradation	Lecture	AI-supported degradation modelling	Explain degradation models
	32	Estimating the degradation function	Lecture + problem solving	AI-assisted estimation examples	Understand degradation estimation
	33	Inverse filtering	Lecture + demo	AI-supported restoration comparisons	Apply inverse filtering concepts
12	34	Minimum mean square error filtering	Lecture + examples	AI-assisted MMSE interpretation	Understand MMSE filtering
	35	Constrained least square filtering	Lecture + discussion	AI-based restoration parameter tuning	Analyze constrained least square restoration
	36	Geometric mean filter	Lecture + coding demo	AI-supported filter comparison	Apply geometric mean filtering
13	37	Biomedical image restoration case studies	Case study + discussion	AI-generated restoration insights	Relate restoration methods to diagnosis
	38	Biomedical image reconstruction workflows	Lecture + examples	AI-assisted reconstruction planning	Understand image reconstruction strategies
	39	Integrated review of image enhancement and restoration	Discussion + examples	AI-based topic summarization	Consolidate restoration and enhancement concepts
14	40	AI in biomedical image analysis	Lecture + case study	AI case studies	Understand AI applications in biomedical imaging
	41	Recent trends in medical image processing	Lecture + discussion	AI-supported literature exploration	Identify emerging developments
	42	Course review and project discussion	Discussion	AI-assisted summary tools	Consolidate course understanding
15	43	Case Study of Unit I and II	Discussion	AI-supported case interpretation	Review biomedical imaging modalities
	44	Case Study of Unit III and IV	Discussion	AI-assisted image	Review quality analysis

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
				quality and Fourier analysis	and spectral methods
	45	Case Study of Unit V	Discussion	AI-based restoration evaluation	Consolidate restoration and reconstruction

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DISCRETE MATHEMATICS & GRAPH THEORY

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2303	**Course Type: Integrated** Program Core Course
Programs Offered: CSE, CSME	**Total Hours:** 45 Hours
Semester: III	**Hours/Week:** 03 Hours
Credits: 03	**L – T – P – J:** 3 – 0 –0 – 0
Prerequisites: Basic Mathematics – Algebra, Matrix, Modulus	

Course Vision

This course equips students with rigorous discrete mathematical foundations essential for computer science and engineering. Through logical reasoning, graph theory, and algebraic structures, students develop analytical problem-solving skills applicable to real-world computational systems. The curriculum bridges theoretical mathematics with practical applications in networking, cryptography, and algorithm design.

Course Objectives

Students will:

1. Master the art of formal reasoning and proof construction to ensure the structural integrity and correctness of computational logic.
2. Comprehend and manipulate the foundational elements of discrete systems, including sets, relations, and functions, to organize and manage complex data.
3. Gain a deep understanding of graph theory to model, visualize, and optimize interconnected systems and hierarchical data structures.
4. Develop the ability to quantify uncertainty and evaluate risk within digital environments using statistical tools and discrete probability distributions.

Course Outcomes (CO):

At the end of the course students will be able to:

CO	Outcome	Bloom taxonomy level
CO1	Understand and construct precise mathematical proofs	L2
CO2	Understand the basic principles of sets and operations in sets and apply counting principles to determine probabilities. Be familiar with recurrence relations.	L2
CO3	Apply graph theory techniques to solve wide variety of real-world problems	L4
CO4	Demonstrate the use of tree algorithms, graph colouring, and planar theorems to solve optimization and spatial arrangement problems.	L5
CO5	Apply discrete probability foundations and random variable distributions to model uncertain events and calculate expected outcomes in stochastic systems.	L3

Broad Course Description

This course offers a comprehensive grounding in Discrete Mathematics and Probability Theory, covering the essential mathematical structures required for computer science and software engineering. Students begin by mastering formal logic, set theory, and combinatorics to build rigorous arguments and optimize algorithmic efficiency, which are critical for database management and digital circuit design. The curriculum then transitions into graph theory and statistical modelling, teaching students how to represent complex networks and apply discrete probability distributions to quantify uncertainty. Ultimately, these modules provide the analytical framework necessary for high-level applications in cryptography, network architecture, and data analysis.

Detailed Course Syllabus

The course consists of five modules. Each module integrates:

- Formal Mathematical Reasoning and Proof Construction
- Structural Modelling using Sets, Relations, and Functions
- Graph-Theoretic Analysis and Network Connectivity
- Planar Graph Properties and Combinatorial Optimization
- Algebraic Structures and Cryptographic Applications

Course Modules

MODULE 1: MATHEMATICAL FOUNDATIONS & LOGIC [06 Hours]

Propositional Logic - Propositions, logical connectives: AND, OR, NOT, IMPLICATION, BICONDITIONAL; Truth tables and tautologies; logical equivalence and laws of logic; Normal forms: CNF (Conjunctive) and DNF (Disjunctive Normal Form).

Applications *in digital circuit design and SAT solvers*

Predicate Logic & Proofs - Predicates, quantifiers (∀, ∃), nested quantifiers; Rules of inference: Modus Ponens, Modus Tollens, Hypothetical Syllogism; Proof techniques: Direct proof, Proof by contradiction, Mathematical induction.

Applications: *First-order logic as the backbone of AI knowledge bases and expert systems*

Combinatorics & Counting - Permutations and combinations; Binomial theorem; Pigeonhole principle and its applications; Recurrence relations: Fibonacci sequence, Towers of Hanoi.

Applications: *Complexity analysis and counting in algorithm design and AI search spaces*

MODULE 2: SET THEORY, RELATIONS & FUNCTIONS [06 Hours]

Set Theory - Sets, subsets, power sets, Cartesian products; Set operations: Union, Intersection, Difference, Complement; Venn diagrams; Principle of Inclusion-Exclusion; infinite sets and cardinality;

Applications: *Set-theoretic operations in query languages (SQL) and AI feature spaces*

Relations - Relations: definition, matrix and digraph representations; Properties: reflexive, symmetric, antisymmetric, transitive; Equivalence relations and partitions; partial orders and Hasse diagrams. ***Applications:*** *Relational DB, ontologies, and semantic AI models built on relation algebra*

Functions - Functions: injective, surjective, bijective; composition and inverse functions; Recursively defined functions and structural induction

Applications: *Activation functions in neural networks*

MODULE 3: ADVANCED ALGEBRAIC STRUCTURES [07 Hours]

Group Theory: Axioms of Groups: Closure, associativity, identity, and inverses. Subgroups and Cosets, Cyclic Groups, Permutation Groups.

Application: *Exploring how group theory forms the basis of the* ***Discrete Logarithm Problem***

Rings and Fields: Ring Fundamentals: Definitions of Rings, Commutative Rings, and Integral Domains.

Ideals and Factor Rings, From Rings to Fields, Finite Fields ($GF(p^n)$).
Application: *The use of Finite Fields in* ***Advanced Encryption Standards (AES)*** *and polynomial-based error detection.*

MODULE 4: GRAPH THEORY — CORE CONCEPTS [05 Hours]
Introduction to Graphs - Definitions: vertices, edges, degree, handshaking lemma; Types: undirected, directed (digraphs), weighted, multigraphs, simple graphs; Graph representations: adjacency matrix, adjacency list, incidence matrix, graph isomorphism. Sub Graph & Spanning subgraph.
Applications: *Graph databases (Neo4j), knowledge graphs in AI systems (Google Knowledge Graph)*
Graph Connectivity & Traversal - Paths, cycles, connectivity; strongly and weakly connected components; Euler paths and circuits (Königsberg problem); Hamiltonian paths and cycles; ***(Applications*** *in digital circuit design and SAT solvers*

MODULE 5: SPECIAL GRAPHS [06 Hours]
Special Graphs and Trees: Complete Graphs: Kn properties, edge formula n(n-1)/2, and complete bipartite structures (Km,n); Regular Graphs: Degree sequences, k-regularity, and existence conditions; Trees: Acyclic characterizations, forests, pendant vertices, and the two-leaf existence theorem.
Applications: Decision trees and random forests in ML; parse trees in NLP; Monte Carlo Tree Search in game AI
Planar Graphs, Coloring & Matching - Planar graphs: Euler's formula (V – E + F = 2), Kuratowski's theorem; Graph coloring: chromatic number, greedy coloring, four-color theorem; Bipartite graphs;
Applications: Register allocation in compilers; scheduling problems; bipartite matching in recommendation systems
CASE STUDIES
CASE STUDY 1: NVIDIA Course Certification: Introduction to Graph Neural Networks (Self-Paced Learning)
Important concepts of graphs, how to apply neural networks to graphs, Various applications of graph neural networks, how to build and efficiently train GNN-based models
Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-05+V1
CASE STUDY 2: Students present a short research review. Students review on at least one recent research topic such as:

- AI-based neural architecture search (NAS) for optimized deep learning models
- Deep learning approaches for medical image classification and segmentation
- Activation functions and loss function optimization using intelligent methods
- Backpropagation and gradient optimization techniques in deep networks
- Convolutional Neural Networks (CNNs) for object detection and pattern recognition
- Transfer learning and fine-tuning strategies for healthcare applications
- Autoencoders and GANs for synthetic medical image generation
- Deep learning for time-series analysis in healthcare monitoring
- Explainable AI (XAI) for clinical decision support systems

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration:**
  - **Module 1:** AI-assisted translation of natural language into formal predicates and automated truth table generation using SageMath, Logisim-evolution, and ChatGPT for logic formalization.
  - **Module 2:** AI-generated visualization of Hasse diagrams and automated mapping of complex set relations using SageMath, Wolfram Alpha, and GeoGebra.
  - **Module 3:** AI-powered conversion of graph matrices into visual digraphs and connectivity visualizations using Gephi, NetworkX (Python), and GraphViz.
  - **Module 4:** AI-assisted simulators for tree traversals and automated greedy coloring of planar graphs using SageMath, Gephi, and VisuAlgo.
  - **Module 5:** AI-generated exploration of group axioms and automated verification of algebraic properties in finite fields using SageMath, Wolfram Alpha, and GAP (Groups, Algorithms, Programming).
- **Authentic Intelligence Reflection:**
  - **Module 1:** AI-generated logical proofs for hallucinations and validate inductive steps manually using Z3 Theorem Prover and Lean Proof Assistant.
  - **Module 2:** AI-rendered relations and identify edge cases in infinite sets using Wolfram Alpha and SageMath outputs.
  - **Module 3:** AI-identified graph isomorphisms and network connectivity paths using NetworkX and Gephi-generated results.
  - **Module 4:** AI-suggested graph colorings and verify tree-search logic using VisuAlgo simulations and SageMath outputs.
  - **Module 5:** AI-generated algebraic proofs and validate finite field arithmetic used in cryptographic protocols using SageMath and GAP verification outputs.

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.

A. Professional Core Courses (Theory)

Criteria	Weightage of Marks
Assessment 1 (Case studies on applications)	10
AI-powered Assignment 2	10
Quiz	10
Mid Semester Examination (MSE)	30
Total	**60 marks**

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**	**3**
CO1	3	2	-	-	-	-	-	-	-	-	2	-	2	2
CO2	3	2	-	-	-	-	-	-	-	-	2	-	3	3
CO3	3	3	2	2	-	-	-	-	-	-	2	-	3	3
CO4	3	3	3	2	2	-	-	-	-	-	2	-	1	2
CO5	3	3	2	2	-	-	-	-	-	-	2	-	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Recommended Tools

1. **SageMath (The All-In-One Solution)** - SageMath is an open-source alternative to Magma or Maple. It is built on Python, it allows students to programmatically verify graph isomorphisms, calculate chromatic numbers, and simulate complex probability distributions in a single environment.
2. **Logisim-evolution (Logic & Circuits)** - This is a modern, educational tool for designing and simulating digital logic circuits.
3. **Gephi (Graph & Network Visualization)** - Gephi is the leading open-source platform for visualizing and exploring large-scale networks.
4. **Z3 Theorem Prover (Formal Verification)**- Developed by Microsoft Research, Z3 is a state-of-the-art "Satisfiability Modulo Theories" (SMT) solver.
5. **GeoGebra (Interactive Probability)** - GeoGebra is a dynamic mathematics software that excels at visualizing statistical functions.

Suggested Reading

Primary Textbooks

1. ***Kenneth H. Rosen, Discrete Mathematics and Its Applications,*** *8th Edition, McGraw-Hill Education, New York, 2018. ISBN: 978-1-259-67651-2.*
2. ***Ralph P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction,*** *5th Edition, Pearson Education, New Delhi, 2013. ISBN: 978-0-201-72634-3.*

Reference Books

1. ***C.L. Liu and D.P. Mohapatra, Elements of Discrete Mathematics,*** *4th Edition, Tata McGraw-Hill Education, New Delhi, 2017. ISBN: 978-0-07-066913-5.*
2. ***J.A. Bondy and U.S.R. Murty, Graph Theory,*** *1st Edition, Springer Graduate Texts in Mathematics, New York, 2008. ISBN: 978-1-846-28969-5.*

3. ***David S. Dummit and Richard M. Foote, Abstract Algebra,*** *3rd Edition, John Wiley & Sons, New Jersey, 2004. ISBN: 978-0-471-43334-7.*
4. ***Susanna S. Epp, Discrete Mathematics with Applications,*** *5th Edition, Cengage Learning, Boston, 2019. ISBN: 978-1-337-69419-4.*

Additional Resources

1. ***Last-Minute Notes:*** *https://www.geeksforgeeks.org/last-minute-notes-engineering-mathematics*
2. **MIT OpenCourseWare:** *https://ocw.mit.edu/courses/6-042j*
3. **Combinatorica (Wolfram Mathematica):**
4. *https://reference.wolfram.com/language/Combinatorica*
5. **Brilliant.org:**
   - *Course Link: https://brilliant.org/courses/discrete-mathematics*
   - *Probability Track: https://brilliant.org/courses/probability-fundamentals*
   - Numberphile (YouTube) : ***https://youtube.com/@numberphile***

AI-Powered Assignments

Assignment 1: AI-Generated Truth Table and Logic Circuit Evaluation

- **Students prompt an AI tool:** *"Generate truth tables and logic expressions for a given Boolean function: F(A,B,C) = A'B + BC' + ABC."*
- **Students must:**
  1. analyze the correctness of the generated truth table
  2. identify errors, missing minterms, or incorrect simplification
  3. improve the solution using Karnaugh maps or Boolean algebra laws
- **Assessment focuses on human judgment, logical reasoning, and understanding of Boolean algebra.**

Assignment 2: AI-Assisted Set Theory and Probability Problem Solving

- **Students prompt an AI tool:** *"Solve a set theory problem involving union, intersection, complement, and Venn diagram representation for three sets."*
- **Students must:**
  1. evaluate the relevance and correctness of the generated solution
  2. identify missing steps or incorrect set operations
  3. present an improved solution with proper set notation and Venn diagram visualization
- **Assessment focuses on creativity, accuracy, and practical usefulness in probability and statistics.**

Assignment 3: AI-Based Proof Validation for Mathematical Induction

- **Students prompt an AI tool:** *"Generate a proof by induction for the formula: $1^2 + 2^2 + 3^2 + ... + n^2 = n(n+1)(2n+1)/6$."*
- **Students must:**
  1. verify whether the proof correctly follows induction steps (base case, inductive hypothesis, inductive step)
  2. identify missing steps or logical gaps
  3. redesign the proof with clear justification and algebraic manipulation
- **Assessment focuses on critical evaluation and understanding of mathematical induction.**

Assignment 4: AI-Supported Recurrence Relation Solution

- **Students prompt an AI tool:** *"Solve the recurrence relation: $T(n) = 2T(n/2) + n$ with $T(1) = 1$ using Master Theorem."*
- **Students must:**

1. assess whether the solution correctly applies Master Theorem conditions
2. identify technical limitations or incorrect **case selection**
3. enhance the solution with step-by-step derivation and complexity analysis

- **Assessment focuses on analytical thinking and recurrence solving skills.**

Assignment 5: AI-Generated Graph Algorithm Workflow Review

- **Students prompt an AI tool:** *"Generate a step-by-step workflow for finding the shortest path in a weighted graph using Dijkstra's algorithm."*
- **Students must:**
  1. analyze the feasibility and correctness of the workflow
  2. identify gaps in graph representation, edge case handling (negative weights, disconnected nodes)
  3. improve the workflow with proper initialization, priority queue usage, and complexity analysis
- **Assessment focuses on practical reasoning and graph algorithm design quality.**

Assignment 6: AI-Assisted Combinatory Problem Solver

- **Students prompt an AI tool:** *"Solve a combinatory problem: How many ways to select a committee of 3 from 10 people with conditions."*
- **Students must:**
  1. evaluate the correctness of permutation/combination formulas applied
  2. identify missing constraints or over counting issues
  3. present an improved solution using proper combinatorial reasoning
- **Assessment focuses on combinatorial thinking and counting principles.**

Mini Project Component

Industry-Driven Projects

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

Project 1 – Boolean Logic Simplifier and Circuit Designer

- **Problem:** Design a tool that simplifies Boolean expressions using Karnaugh maps and generates corresponding logic gate circuits.
- **Concepts used:** Boolean algebra, logic gates, K-map (2,3,4 variable), SOP, POS, NAND/NOR implementation
- **Industry relevance:** Digital circuit design, FPGA programming, Computer architecture

Project 2 – Set Theory-Based Database Query Optimizer

- **Problem:** Implement a query optimizer that uses set operations (union, intersection, difference) to optimize database queries.
- **Concepts used:** Set theory, relations, Cartesian product, joins, query optimization
- **Industry relevance:** Database management systems (MySQL, PostgreSQL), Query processing

Project 3 – Mathematical Induction Prover

- **Problem:** Build a tool that validates mathematical induction proofs for summation formulas, inequalities, and divisibility problems.
- **Concepts used:** Principle of mathematical induction, strong induction, well-ordering principle
- **Industry relevance:** Algorithm correctness verification, Formal methods

Project 4 – Recurrence Relation Solver and Complexity Analyzer

- **Problem:** Develop a system that solves recurrence relations using Master Theorem, substitution method, and recursion tree method, and analyzes time complexity.

- **Concepts used:** Recurrence relations, Master Theorem, divide and conquer, complexity analysis (Big O, Theta, Omega)
- **Industry relevance:** Algorithm analysis, Performance optimization

Project 5 – Permutation and Combination Calculator

- **Problem:** Design a combinatorial calculator that computes permutations, combinations, and arrangements with constraints.
- **Concepts used:** Fundamental counting principle, permutations, combinations, binomial theorem, pigeonhole principle
- **Industry relevance:** Probability and statistics, Cryptography, Game theory

Project 6 – Graph Theory-Based Social Network Analyzer

- **Problem:** Implement a social network analysis tool that finds shortest paths, influential nodes, and communities using graph algorithms.
- **Concepts used:** Graph representation (adjacency matrix/list), BFS, DFS, shortest path, centrality measures, connected components
- **Industry relevance:** Social media (Facebook, LinkedIn), Network analysis

Project 7 – Shortest Path Visualizer for Road Networks

- **Problem:** Develop an interactive tool to find and visualize shortest paths in road networks using Dijkstra's and A* algorithms.
- **Concepts used:** Weighted graphs, Dijkstra's algorithm, A* algorithm, priority queue, path visualization
- **Industry relevance:** GPS navigation (Google Maps, Uber), Logistics optimization

Project 8 – Minimum Spanning Tree for Network Design

- **Problem:** Design a tool that finds minimum spanning tree for connecting nodes in a telecommunication or power grid network.
- **Concepts used:** Spanning trees, Prim's algorithm, Kruskal's algorithm, Union-Find data structure
- **Industry relevance:** Telecom infrastructure, Power grid planning, Cable TV networks

Project 9 – Graph Colouring for Exam Scheduling

- **Problem:** Implement a graph coloring algorithm to solve exam scheduling problems (timetabling) by assigning time slots without conflicts.
- **Concepts used:** Graph coloring, chromatic number, greedy coloring, backtracking
- **Industry relevance:** University timetabling, Resource allocation, Register allocation in compilers

Project 10 – Eulerian and Hamiltonian Path Finder

- **Problem:** Build a tool that detects Eulerian and Hamiltonian paths/cycles in graphs for route planning applications.
- **Concepts used:** Eulerian paths/circuits, Hamiltonian paths/cycles, Fleury's algorithm, backtracking
- **Industry relevance:** Route inspection (postman problem), Traveling salesman problem (TSP)

Department of Computer Science & Medical Engineering

Weekly Lecture Hours

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Discrete Mathematics, Propositions and Logical Connectives	Lecture + Examples	AI-assisted truth table generation	Understand the importance of discrete mathematics and propositional logic
	2	Truth Tables, Tautologies, Logical Equivalence, Laws of Logic	Lecture + Problem Solving	AI-generated logical reasoning examples	Construct and simplify logical expressions
	3	CNF and DNF (Normal Forms)	Lecture + Discussion	AI-assisted SAT solver demonstration	Convert logical expressions into standard normal forms
2	4	Predicate Logic, Predicates and Quantifiers	Lecture + Examples	AI knowledge-base representation	Apply predicate logic in AI reasoning
	5	Rules of Inference (Modus Ponens, Modus Tollens, Hypothetical Syllogism)	Lecture + Exercises	AI-based inference engine examples	Apply inference rules to logical reasoning
	6	Proof Techniques: Direct Proof, Contradiction, Mathematical Induction	Lecture + Problem Solving	AI-generated proof verification	Develop mathematical proof skills
3	7	Permutations and Combinations	Lecture + Examples	AI-assisted combinatorial calculations	Solve counting problems
	8	Binomial Theorem and Pigeonhole Principle	Lecture + Discussion	AI-generated counting scenarios	Apply counting principles to computational problems
	9	Recurrence Relations: Fibonacci and Towers of Hanoi	Lecture + Demonstration	AI visualization of recursive algorithms	Analyze recursive mathematical models
4	10	Sets, Subsets, Power Sets and Cartesian Products	Lecture + Examples	AI-generated set visualizations	Understand fundamental set operations

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	11	Union, Intersection, Difference, Complement, Venn Diagrams	Lecture + Exercises	AI-assisted Venn diagram generation	Perform operations on sets
	12	Inclusion-Exclusion Principle, Infinite Sets and Cardinality	Lecture + Discussion	AI-assisted dataset analysis	Apply counting techniques using set theory
5	13	Relations and their Matrix Representation	Lecture + Examples	AI-supported relation visualization	Represent relations mathematically
	14	Digraph Representation and Properties of Relations	Lecture + Problem Solving	AI-generated graph representations	Analyze properties of relations
	15	Equivalence Relations, Partitions, Partial Orders and Hasse Diagrams	Lecture + Discussion	AI ontology representation	Model hierarchical relationships
6	16	Functions: Injective, Surjective and Bijective	Lecture + Examples	AI-assisted function mapping	Classify mathematical functions
	17	Composition and Inverse Functions	Lecture + Exercises	AI-based function visualization	Solve problems involving function composition
	18	Recursive Functions and Structural Induction	Lecture + Discussion	AI recursive reasoning examples	Apply recursion in mathematical models
7	19	Group Theory: Definition and Group Axioms	Lecture + Examples	AI-supported algebra visualization	Understand algebraic structures
	20	Subgroups, Cosets and Cyclic Groups	Lecture + Problem Solving	AI-generated cyclic group examples	Analyze subgroup structures
	21	Permutation Groups and Cryptographic Applications	Lecture + Discussion	AI-assisted cryptography demonstration	Relate group theory to security applications

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
8	22	Rings: Definitions, Commutative Rings and Integral Domains	Lecture + Examples	AI-generated algebraic structures	Understand ring theory fundamentals
	23	Ideals, Factor Rings and Fields	Lecture + Discussion	AI-supported finite field visualization	Explain field structures
	24	Finite Fields ($GF(p^n)$) and AES Applications	Lecture + Case Study	AI-assisted AES encryption demonstration	Apply finite fields in cryptography
9	25	Introduction to Graphs, Vertices, Edges, Degree and Handshaking Lemma	Lecture + Examples	AI-generated graph visualization	Understand graph fundamentals
	26	Types of Graphs and Graph Representations	Lecture + Exercises	AI-assisted graph database examples	Represent graphs efficiently
	27	Graph Isomorphism, Subgraphs and Spanning Subgraphs	Lecture + Discussion	AI-supported graph matching	Analyze graph equivalence
10	28	Paths, Cycles and Graph Connectivity	Lecture + Examples	AI-assisted path finding	Explain graph connectivity
	29	Strongly and Weakly Connected Components	Lecture + Discussion	AI-generated network analysis	Analyze directed graphs
	30	Euler and Hamiltonian Graphs	Lecture + Problem Solving	AI-assisted route optimization	Solve Eulerian and Hamiltonian problems
11	31	Complete Graphs and Complete Bipartite Graphs	Lecture + Examples	AI graph analytics	Analyze special graph structures
	32	Regular Graphs and Degree Sequences	Lecture + Exercises	AI-assisted graph property analysis	Identify regular graph properties
	33	Trees, Forests and Pendant Vertices	Lecture + Demonstration	AI-generated tree structures	Understand tree-based data structures
12	34	Tree Properties and Two-Leaf Theorem	Lecture + Discussion	AI-assisted decision tree visualization	Explain tree characteristics
	35	Decision Trees, Random Forests and Parse Trees	Lecture + Case Study	AI model visualization	Relate trees to machine learning

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	36	Monte Carlo Tree Search Applications	Lecture + Demonstration	AI gameplay examples	Apply tree search concepts
13	37	Planar Graphs and Euler's Formula	Lecture + Examples	AI-assisted planar graph visualization	Understand planar graph properties
	38	Kuratowski's Theorem	Lecture + Discussion	AI-generated graph examples	Identify planar and non-planar graphs
	39	Graph Coloring and Chromatic Number	Lecture + Exercises	AI scheduling demonstration	Apply graph coloring algorithms
14	40	Greedy Coloring, Four Color Theorem	Lecture + Discussion	AI-assisted compiler register allocation	Solve graph coloring problems
	41	Bipartite Graphs and Matching	Lecture + Examples	AI recommendation system examples	Apply bipartite matching concepts
	42	Applications in Scheduling, Recommendation Systems and Knowledge Graphs	Lecture + Case Study	AI-enabled graph analytics	Relate graph theory to AI applications
15	43	Case Study: Logic, Set Theory and Algebraic Structures	Discussion	AI-assisted concept summarization	Integrate concepts from Modules I–III
	44	Case Study: Graph Theory and Trees	Discussion + Problem Solving	AI-generated graph problem solving	Integrate graph theory concepts
	45	Comprehensive Course Review and Emerging AI Applications	Interactive Discussion	AI-assisted revision and assessment	Consolidate complete course outcomes

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DATABASE MANAGEMENT SYSTEMS

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

Course Code:25SM2304	**Course Type:** Program Core Course
Programs Offered: CSE, AI and ML, DS, CY, CST, CSME, AI and DS, AI & Robotics	**Total Hours:** 45 (L) + 30 (P)
Semester: III	**Hours/Week:** 05 Hours
Credits: 04	**L – T – P – J:** 3 – 0 – 2 – 0
Prerequisites: Programming Fundamentals, Data Structures	

Course Vision:

This course builds strong foundational knowledge in database design, relational models, SQL query language, and transaction management, while integrating AI-assisted tools and intelligent querying to prepare students for modern data-driven engineering challenges.

Course Objectives:

Students will:

1. Acquire the concept of databases and the Relational Model, Entity-Relationship Model, and design databases for real-world scenarios.
2. Develop queries to extract and manipulate data from databases using Structured Query Language (SQL).
3. Draw and interpret ER Diagrams including entity sets, relationship sets, complex attributes, mapping cardinalities, and weak entities; optimize database design using Normalization concepts.
4. Understand the importance of Transaction Management, Concurrency Control mechanisms, and Recovery Techniques.
5. To equip students with the technical skills to build scalable data infrastructures by mastering the transition from traditional SQL feature engineering to modern AI-driven workflows, including vector search and natural language-to-SQL integration.

Course Outcomes (CO)

CO	Course Outcome	Bloom taxonomy level
CO1	Understand the basic concepts of database management system and Relational Model and Create E-R diagrams and design relational schema.	L2
CO2	Utilize SQL concepts to build and manipulate relational databases for a given schema.	L3
CO3	Design document-based database by applying NoSQL principles and apply normalization techniques in designing the relational database.	L3
CO4	Understand the Transaction Management, concurrency control and recovery management techniques.	L2
CO5	Design a scalable data pipeline that leverage SQL feature engineering and vector embeddings to build AI-driven systems capable of natural language querying.	L3

Department of Computer Science & Medical Engineering

Broad Course Description

Database Management Systems (DBMS) is a core computing course that develops strong foundations in the design, implementation, and management of modern database systems. The course covers essential concepts such as data models, ER modelling, relational schema design, normalization, SQL programming, query processing, transaction management, concurrency control, and recovery mechanisms. Students also explore contemporary database architectures, including distributed and cloud-based systems, with emphasis on efficient data storage, retrieval, and management.

Through laboratory sessions and practical assignments, students gain hands-on experience in SQL DDL/DML operations, indexing, views, stored procedures, and triggers. The course further introduces NoSQL technologies such as MongoDB for document-oriented database design and CRUD operations. In alignment with emerging industry trends, it also covers AI-driven database applications including natural language-to-SQL translation, intelligent querying, semantic search, and vector databases, preparing students for advanced careers in data engineering and analytics.

Detailed Course Syllabus

Module – I: Relational Databases & ER Diagrams **09 Hours**

Purpose of Database System – Views of Data – Data Models – Database System Architecture – Database System Applications – Introduction to Relational Databases – Structure of Relational Databases – Database Schema – Keys – Schema Diagrams – Entity-Relationship Model – E-R Diagrams – Types of Attributes (Simple, Composite, Multivalued, Derived) – Entity Sets and Relationship Sets – Mapping Cardinalities (One-to-One, One-to-Many, Many-to-Many) – Participation Constraints – Weak Entity Sets – Extended ER Features – ER-to-Relational Schema Mapping – Features of Good Relational Designs.

Engineering Applications: Enterprise data management, healthcare record systems, banking and financial databases.

Text Book 1: Chapter 1, 2,6

Module – II: Relational Query Language **09 Hours**

Overview of the SQL Query Language – SQL Data Definition – Basic Structure of SQL Queries – Additional Basic Operations – Aggregate Functions – Nested Subqueries – Join Expressions – Views – Transactions – SQL Data Types and Schemas – Authorization – Accessing SQL from a Programming Language – Functions and Procedures – Triggers.

Engineering Applications: Data extraction in business intelligence, dynamic report generation, web application backends.

Text Book 1: Chapter 3, 4, 5

Module – III: Database Design & NoSQL **10 Hours**

Part A – Normalization: Decomposition Using Functional Dependencies – Non-loss Decomposition – First, Second, Third Normal Forms – Dependency Preservation – Boyce-Codd Normal Form – Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form.

Part B – NoSQL & MongoDB: Introduction to NoSQL & MongoDB Architecture – NoSQL vs. RDBMS, CAP Theorem, ACID vs. BASE – MongoDB Data Model: Databases, Collections, Documents – Core CRUD Operations: insertOne, insertMany, find with Projections, Comparison/Logical Operators, updateOne, updateMany, deleteOne, deleteMany – Aggregation Pipeline Basics.

Engineering Applications: Hospital management systems, e-commerce databases, academic information systems.

Text Book 1: Chapter 7 (Normalization); Text Book 3: Chapters 1, 2, 3, 4 (MongoDB & NoSQL)

Module – IV: Transaction Management **09 Hours**

Transaction Concepts – ACID Properties – A Simple Transaction Model – Transaction Atomicity and Durability – Transaction Isolation – Schedules – Serializability – Transactions as SQL Statements – Concurrency Control – Lock Based Protocols – Deadlock Handling – Recovery System – Failure Classification – Recovery Algorithm – Buffer Management – ARIES.

Engineering Applications: Banking systems, airline reservation, stock trading platforms requiring ACID guarantees.

Text Book 1: Chapter 17, 18, 19

Module – V: SQL Meets AI – Integrating to Future **08Hours**

Preparing Data for Machine Learning – Data Cleaning & Normalization – Feature Engineering with SQL – Vector Search & Embeddings – Scalable Pipelines – AI-Assisted SQL & Intelligent Querying – Natural Language to SQL Translation.

Engineering Applications: AI-powered recommendation engines, semantic search systems, large-scale ML data pipelines.

Text Book 2: Chapter 24

CASE STUDY 1: NVIDIA Course Certification: Use GPUs to Accelerate RAG Pipelines with Vector Databases (Self-Paced Learning)

Learn the differences between vector databases and traditional databases such as relational and nosql, become familiar with the benefits and drawbacks to the different types of vector database offerings available and how to best utilize them for their semantic search workloads, learn how to optimize your vector database workloads for high-speed index creation at scale, learn how to select the best indexes for your data needs, and how and when GPUs can provide the best benefit

Link:https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-08+V1

CASE STUDY 2: Students present a short research review. Students review on *at least one recent research topics* such as:

- AI-powered automatic SQL query optimization and tuning
- Machine learning for adaptive indexing in relational databases
- Block chain-integrated database systems for data integrity and security
- Distributed NoSQL databases for cloud-native applications
- Graph neural networks for knowledge graph construction and querying
- Automated database design using deep learning and reinforcement learning
- AI-driven anomaly detection in transaction processing systems

CASE STUDY 3: Problem-First Learning (Instead of Topic-First)

- Start with **"How does Amazon manage millions of products and customer orders simultaneously?"**, then teach **Relational Databases, Schemas, and Keys**, *instead of*
- Today we learn Relational Databases and then linking to application Amazon.
- Start with **"How does Flipkart/Amazon retrieve product details using a product ID?"**, then teach **SQL SELECT Queries and Primary Keys**.
- Start with **"How does a banking system ensure your money transfer is not lost or duplicated?"**, then teach **Transactions and ACID Properties**.
- Start with **"How does Facebook store and query billions of user connections efficiently?"**, then teach **NoSQL Graph Databases and Neo4j**.
- Start with **"How does a hospital manage patient records, doctors, appointments, and billing without data redundancy?"**, then teach **Normalization (1NF to 3NF)**.

- Start with **"How does Google handle millions of searches per second?"**, then teach **Database Indexing and Query Optimization**.
- Start with **"What happens if two people book the last train ticket at the exact same time?"**, then teach **Concurrency Control and Lock Based Protocols**.

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

AI Integration:

1. Module I: AI-assisted schema design, automated data model generation, AI-based ER diagram creation: ERBuilder Data Modeler
2. Module II: Natural language-to-SQL translation tools: ChatGPT
3. Module III: AI-based normalization validation, AI-assisted NoSQL schema generation and query creation for MongoDB: MongoDB Atlas
4. Module IV: AI-assisted transaction monitoring, anomaly detection in concurrency control scenarios: Splunk
5. Module V: AI-powered vector search, large language model integration with SQL pipelines: Pinecone

Authentic Intelligence Reflection:

1. **Module I:** Evaluate the reliability of AI-generated database schemas and understand the limitations of automation in database design.
2. **Module II:** Critically assess the accuracy of natural language-to-SQL translations and verify query correctness.
3. **Module III:** Analyse the trade-offs in AI-suggested normalization and discuss the implications for performance versus consistency. Students also evaluate when NoSQL is preferable to RDBMS and critically examine CAP Theorem trade-offs in distributed data systems.
4. **Module IV:** Evaluate the ethical and safety implications of automated transaction recovery in critical systems.
5. **Module V:** Reflect on fairness, privacy, and transparency when integrating AI with large-scale data systems.

Laboratory Component

The laboratory emphasises hands-on implementation, SQL programming, and AI-assisted debugging exercises using Mysql, MongoDB.

Lab Experiments:

1. Database Table Creation and Manipulation – Consider the schema for UNIVERSITY Database:
   STUDENT (USN, SName, Address, Phone, Gender)
   SEMSEC (SSID, Sem, Sec)
   CLASS (USN, SSID)
2. Consider the schema for UNIVERSITY Database:
   STUDENT (USN, SName, Address, Phone, Gender)
   SEMSEC (SSID, Sem, Sec)
   CLASS (USN, SSID)
   SUBJECT (Subcode, Title, Sem, Credits)
   IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)
3. Write SQL queries to:
   1. Identify 5 Fast learners considering performance of both Test 1 & Test2 combined, Modify the query to retrieve 5 slow learners.
   2. Calculate the FinalIA (average of best of two test marks) and update the corresponding table for all

students.

3. Categorize students based on the following criterion If finalia=17 to 20 then cat='Outstanding' If finalia=12 to 16 then cat='Average' If finalia<12 then cat='weak' Give these details only for 8th semester A, B, and C section students.

4. Write SQL queries to:
   1. Count the number of students whose final IA marks is above SUBJECT average FOR DBMS.
   2. Find the total count of Credits for the given student 'avalewis'.
   3. Classify all FEMALE and MALE students who have opted for 'Digital Logic' using UNION.
5. Write SQL queries to:
   - Retrieve Students and Their Semester Details using NATURAL JOIN.
   - Retrieve Student Marks with Subject Details using Equijoin.
   - Retrieve All Students and Their Marks (Including Students Without Marks) using Left Outer Join.
   - Retrieve All Subjects and Their Students (Including Subjects Without Students) using Right Outer Join.
   - Retrieve All Subjects and Their Students (Including Subjects Without Students) using FULL Outer Join.
6. Calculate the Grade based on final IA marks and update Grade column of the corresponding table for all students using user defined function and stored procedure.
   and Transaction Control Language (TCL) commands.
7. Write a SQL transaction to insert a new student into the STUDENT table and enroll the student in a specific class in the CLASS table. If any error occurs during the enrollment, roll back the transaction.
   Create a transaction with a save point. If an error occurs after the save point, roll back to the sav epoint without affecting the earlier operations (such as inserting the student record).
8. Write a SQL trigger that automatically logs every insertion into the STUDENT table. The log should contain the action type (INSERT), USN, SName, and the date/time when the insertion occurred. Views – Create views to enhance manageability and data abstraction.
   Create a SQL trigger that logs the details of any student who is deleted from the STUDENT table. The log should include the Action Type (DELETE), USN, SName, and the date/time of the deletion.
   Write a SQL trigger that logs every update made to the name of a student in the STUDENT table. The log should capture USN, SName, and the date/time when the update occurred.
9. Create a view to display basic student information including USN, SName, and Phone from the STUDENT table. Create a view to show students along with their enrolled classes, including USN, SName, SSID (class ID), and Sem (semester). This will involve a join between the STUDENT, CLASS, and SEMSEC tables.
   Create a view to show the total number of students in each semester, grouping by Sem from the SEMSEC table. This will involve the CLASS table to count the number of students per semester.
   Create a view to show only the students who have scored above 15 in Test1 in the IAMARKS table. Include the USN, SName, and Test1 score.
10. Write and execute program to perform operations on MongoDb Database
11. A student performance dataset contains missing values, inconsistent department names, and unscaled marks. Clean the data by handling missing values and standardizing text fields. Normalize the marks column using Min-Max normalization.

Open-Ended Experiments:

1. GUI-Based Database Application – Develop a simple GUI-based database application incorporating table creation, constraints, queries, joins, functions, procedures, transactions, and triggers.
2. Case Study: Real-Life Database Application – Design and implement a database solution for a real-world problem domain like:
   - Student Placement Management System
   - Online Food Ordering System

- E-Commerce Product and Sales Management System

CO – PO / PSO Mapping

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**	**3**
CO1	3	2	2	–	1	–	–	–	–	–	–	3	2	1
CO2	3	3	3	1	3	–	–	–	1	1	–	3	2	1
CO3	3	3	3	2	3	–	–	–	1	–	–	3	3	2
CO4	3	3	3	3	3	1	–	1	1	1	1	3	3	2
CO5	3	3	3	3	3	1	1	1	1	1	1	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Assessment Scheme

Integrated Professional Core Course (Theory + Practical)

Criteria	**Weightage (Marks)**
Continuous Assessment Tool (CAT) every 2 weeks (Min. 06 assessments including AI-powered Assignment)	20
Mid Semester Examination (MSE)	20
Lab Record	05
Lab Examination	15
Total (Internal Assessment)	**60 marks**
Semester End Examination (SEE) – External Assessment	40 marks

Teaching-Learning Process

Pedagogical methods adopted to accelerate the attainment of course outcomes:

1. Lecture Method – Traditional lectures complemented by varied teaching methods to develop course outcomes.
2. Interactive Teaching – Active learning through brainstorming,focused listening.
3. Video/Animation – Films to explain the functioning of database concepts and system architectures.
4. Problem-Based Learning – Analytical and generalisation exercises drawn from real data scenarios.
5. Activity-Based Learning –real-world problem solving and programming demonstrations.

Recommended Tools

1. Database: MySQL
2. Programming Interface: Python
3. AI Tools: GitHubCopilot, ChatGPT / AI SQL assistants, Text-to-SQL tools, Splunk, Pinecone
4. Visualization: ER diagram tools (draw.io, dbdiagram.io), SQL query visualizers

Suggested Reading

Primary Textbooks:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan – Database System Concepts, Seventh Edition, McGraw-Hill, 2020.
2. Daiki Moriyama - SQL for the AI Era: The Complete Handbook for Intelligent Data Systems, Machine Learning Readiness, and Real-World Automation, Richa Publishing Minds, 2025.

3. Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, MongoDB: The Definitive Guide, 3rd Edition, O'Reilly Media.

Reference Books:

1. RamezElmasri, Shamkant B. Navathe – **Fundamentals of Database Systems**, Seventh Edition, Pearson Education, 2017.
2. Raghu Ramakrishnan and Johannes Gehrke – **Database Management Systems**, Third Edition, McGraw-Hill, 2003.
3. C.J. Date, A. Kannan, S. Swamynatham – **An Introduction to Database Systems**, Eighth Edition, Pearson Education, 2012.

E-Resources:

1. IBM Documentation – What is a Database Management System (DBMS)? https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system
2. Oracle – What Is a Database? https://www.oracle.com/in/database/what-is-database/
3. SWAYAM Course – Database Management Systems (CEC Course Preview) https://onlinecourses.swayam2.ac.in/cec19_cs05/preview
4. NPTEL Course – Database Management Systems (Online Certification Course) https://onlinecourses.nptel.ac.in/noc19_cs46/preview
5. arXiv Research Paper – A Study on Database Management Systems (DBMS) https://arxiv.org/pdf/2203.07376
6. Pinecone Docs – Vector Database: Overview and Getting Started Guide https://docs.pinecone.io/guides/get-started/overview

Weekly 45 Hours Lectures

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Database Systems	Lecture + examples	AI-generated database application examples	Understand the purpose and applications of DBMS
	2	Views of Data and Data Models	Lecture + discussion	AI-assisted schema visualization	Understand different data models and views
	3	Database Architecture and Applications	Lecture + case study	AI-based enterprise database examples	Analyze database system architecture
2	4	Introduction to Relational Databases	Lecture + examples	AI-generated relational model examples	Understand relational database concepts
	5	Database Schema and Keys	Lecture + problem solving	AI-assisted key identification	Identify keys and schema structures
	6	Schema Diagrams and Relational Design	Lecture + coding demo	AI-assisted schema mapping	Design relational schemas effectively
3	7	Entity Relationship Model	Lecture + examples	AI-generated ER diagrams	Understand ER modeling concepts
	8	Entity Sets, Relationship Sets, Attributes	Lecture + discussion	AI-assisted attribute classification	Analyze entity and relationship structures

	9	Mapping Cardinalities and Participation Constraints	Lecture + problem solving	AI-generated cardinality examples	Apply cardinality constraints in database design
4	10	Weak Entity Sets and Extended ER Features	Lecture + examples	AI-assisted ER enhancement	Understand weak entities and advanced ER features
	11	ER-to-Relational Mapping	Lecture + coding demo	AI-generated relational conversion	Convert ER models into relational schemas
	12	Features of Good Relational Designs	Lecture + discussion	AI-assisted design evaluation	Evaluate relational database designs
5	13	Introduction to SQL Query Language	Lecture + examples	AI-assisted SQL query generation	Understand SQL basics
	14	SQL Data Definition Language (DDL)	Lecture + coding demo	AI-generated table creation scripts	Create database schemas using SQL
	15	Basic SQL Queries and Operations	Lecture + problem solving	AI-assisted query execution	Write and execute basic SQL queries
6	16	Aggregate Functions and Nested Queries	Lecture + examples	AI-generated aggregation examples	Apply aggregate functions and subqueries
	17	Join Expressions and Views	Lecture + coding demo	AI-assisted join visualization	Implement joins and views
	18	Transactions and SQL Data Types	Lecture + discussion	AI-generated transaction examples	Understand SQL transactions and data types
7	19	Authorization and Security in SQL	Lecture + examples	AI-assisted privilege management	Understand database authorization
	20	SQL Functions and Procedures	Lecture + coding demo	AI-generated stored procedures	Develop SQL functions and procedures
	21	Triggers and Applications	Lecture + problem solving	AI-assisted trigger generation	Implement triggers in databases
8	22	Functional Dependencies and Decomposition	Lecture + examples	AI-assisted dependency analysis	Analyze functional dependencies
	23	First, Second, and Third Normal Forms	Lecture + discussion	AI-generated normalization examples	Apply normalization techniques
	24	Boyce-Codd Normal Form (BCNF)	Lecture + problem solving	AI-assisted BCNF decomposition	Normalize databases using BCNF
9	25	Fourth and Fifth Normal Forms	Lecture + examples	AI-generated dependency mapping	Understand advanced normalization forms
	26	Introduction to NoSQL Databases	Lecture + discussion	AI-assisted NoSQL comparison	Compare NoSQL and relational databases

	27	MongoDB Architecture and Data Model	Lecture + coding demo	AI-generated MongoDB collections	Understand MongoDB architecture
10	28	CRUD Operations in MongoDB	Lecture + coding demo	AI-assisted query generation	Perform CRUD operations in MongoDB
	29	Aggregation Pipeline Basics	Lecture + examples	AI-assisted aggregation analysis	Apply aggregation pipelines
	30	CAP Theorem and ACID vs BASE	Lecture + discussion	AI-generated distributed DB examples	Analyze distributed database principles
11	31	Transaction Concepts and ACID Properties	Lecture + examples	AI-assisted transaction visualization	Understand transaction management
	32	Transaction Isolation and Schedules	Lecture + problem solving	AI-generated schedule analysis	Analyze transaction schedules
	33	Serializability and Concurrency Control	Lecture + coding demo	AI-assisted concurrency simulation	Apply serializability concepts
12	34	Lock-Based Protocols	Lecture + examples	AI-assisted lock management	Understand lock-based concurrency control
	35	Deadlock Handling Techniques	Lecture + discussion	AI-generated deadlock detection	Analyze deadlock handling methods
	36	Recovery Systems and Failure Classification	Lecture + problem solving	AI-assisted recovery simulation	Understand database recovery concepts
13	37	Recovery Algorithms and Buffer Management	Lecture + examples	AI-assisted recovery process visualization	Apply recovery algorithms
	38	ARIES Recovery Algorithm	Lecture + coding demo	AI-generated ARIES workflow	Analyze ARIES recovery mechanism
	39	Data Cleaning and Normalization for AI	Lecture + case study	AI-assisted preprocessing	Prepare datasets for machine learning
14	40	Feature Engineering using SQL	Lecture + coding demo	AI-generated feature extraction	Apply SQL for feature engineering
	41	Vector Search and Embeddings	Lecture + examples	AI-assisted semantic search	Understand vector databases and embeddings
	42	AI-Assisted SQL and Intelligent Querying	Lecture + discussion	Natural language to SQL translation	Analyze AI-driven querying systems
15	43	Scalable ML Data Pipelines	Lecture + case study	AI-assisted pipeline optimization	Understand scalable AI pipelines
	44	AI-Powered Recommendation Systems	Lecture + examples	AI-generated recommendation models	Apply databases in AI recommendation systems

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DATA STRUCTURES AND ALGORITHMS

[As per Choice based credit system (CBCS) Scheme]

Course Code: 25SM2305	**Course Type:** Program Core Course
Programs Offered: CSE, CSME, AI & Robotics	**Total Hours:** 45 (L) + 30 (P)
Semester: III	**Hours/Week:** 05 Hours
Credits: 04	**L – T – P – J:** 3 – 0 – 2 – 0
Prerequisites: Introduction to C Programming	

Course Vision

The course on Data Structures and Algorithms aims to build a strong foundation in computational thinking and problem-solving by equipping students with essential techniques to design, analyse, and optimize algorithms. It focuses on understanding how data is organized, processed, and efficiently utilized to solve real-world problems.

Course Objectives

At the end of course, students will able to:

1. To understand the fundamental concepts of algorithms, their characteristics, and performance evaluation using asymptotic notations.
2. To impart knowledge on mathematical techniques to analyze both recursive and non-recursive algorithms, including solving recurrence relations.
3. To familiarize the basic data structures such as arrays, linked lists, stacks, queues, and trees using C programming concepts.
4. To impart the basic algorithm design techniques such as brute force, divide and conquer, greedy methods, dynamic programming, and backtracking to solve computational problems.
5. To understand and compare different algorithms based on efficiency, scalability, and suitability for real-world applications.

Course Outcomes (CO)

Students can able to achieve:

CO	Course Outcomes	Bloom taxonomy level
CO1	Understand the efficiency analysis of algorithms using asymptotic notations and solve recurrence relations for both recursive and non-recursive algorithms.	L3
CO2	Implement and manipulate basic data structures such as linked lists using C concepts like pointers and dynamic memory allocation. Design and analyze algorithms divide and conquer technique.	L3
CO3	Apply stack, queue, and tree data structures to solve problems involving expression evaluation, traversal, and hierarchical data representation including BST and AVL trees for efficient data organization	L3
CO4	Design and analyze algorithms using brute force, graph traversal, and hashing techniques	L3
CO5	Apply advanced algorithmic techniques such as greedy methods, dynamic programming, and backtracking to solve optimization problems	L3

Broad Course Description

Data Structures and Algorithms (DSA) is a foundational course focused on designing and analyzing efficient computational solutions. It introduces core concepts of algorithms, including their characteristics, performance evaluation, and asymptotic notations, along with mathematical analysis of recursive and non-recursive algorithms. The course also covers essential data structures such as linked lists, stacks, queues, trees and graphs, implemented using C programming concepts like pointers and dynamic memory allocation.

The course emphasizes key algorithm design techniques such as brute force, divide and conquer, greedy methods, dynamic programming, and backtracking. Through practical problem-solving and programming exercises, students learn to evaluate algorithm efficiency, select appropriate data structures, and apply algorithmic thinking to solve real-world computing and engineering problems.

Detailed Course Syllabus

The course consists of five modules. Each module integrates:

- Core topics
- Engineering Applications
- AI Integration
- Authentic Intelligence

Course Modules

Module 1: Foundations of Algorithms **[8 Hours]**

Algorithm: Definition and characteristics - Performance analysis and efficiency - Asymptotic notations: Big-O, Theta, Omega - Mathematical analysis of algorithms: Non-recursive and recursive algorithms - Recurrence relations: Substitution method, recursion tree method, Master theorem.

Module 2: Basic Data Structures **[10 Hours]**

Introduction to Arrays, Pointers and dynamic memory allocation – Divide and Conquer -merge sort and quick sort - Linked lists: Singly and doubly linked lists; operations including insertion, deletion, and traversal, Circular linked list.

Engineering Applications: Music Playlist creation, Textbook Page Number Finder.

Module 3: Stacks, Queues, and Trees **[8 Hours]**

Stacks: Linked list implementation - Queues: Linear queue; linked list implementation - Expression evaluation using stacks - Trees: Binary trees, representation, and traversal techniques - Binary Search Trees (BST): Operations - Balanced trees: AVL trees.

Engineering Applications: Ticket Counter Implementation, Student Details - Effective Storing and Searching.

Module 4: Algorithm Design Techniques **[9 Hours]**

Brute force techniques: String matching - Graph representations: adjacency list, matrix -Graph traversal algorithms: Depth First Search (DFS), Breadth First Search (BFS), Dijkstra's algorithm, Hashing: Open and closed hashing techniques.

Engineering Applications: Delivery Route Optimization, GPS Navigation.

Module 5: Advanced Algorithms **[10 Hours]**

Greedy algorithms: Fractional knapsack problem - Minimum Spanning Trees: Prim's and Kruskal's algorithms - Dynamic programming: Concept and methodology; 0/1 knapsack problem and Floyd-Warshall algorithm - Backtracking: Concept and methodology; Sum of Subsets problem, P, NP, NP Hard, NP Complete.

Engineering Applications: Sales Profit Maximization, Delivery Route Planning, N queens in Chess Board.

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration:**

- Module-2: USF: Data Structure Visualizations https://www.cs.usfca.edu/~galles/visualization/Algorithms.html
- Module-3: Data Structure Visualizer
  - https://dsa-visualizer-delta.vercel.app/visualizer
- Module-4: USF: Data Structure Visualizations
  - https://www.cs.usfca.edu/~galles/visualization/Algorithms.html
- Module-5: National University of Singapore: Visualizing data structures and algorithms through animation
  - https://visualgo.net/en

- **Authentic Intelligence Reflection:**
  - In module 2 to module 4 the visualization tools are used to verify the understating of the core concept that includes linked list, stack, queue, graphs and trees.
  - The above-mentioned visualization tools act as a powerful feedback mechanism to enhance learning. By leveraging such tools, learners can experience adaptive feedback for continuous improvement.

Laboratory Component (1 Credit)

The lab emphasizes **hands-on implementation of data structures and algorithms using C programming language with applicable co-pilot integration, along with visualization and AI-assisted debugging techniques.**

Lab Experiments

1. Sorting algorithms (merge sort, quick sort)
2. Linked list operations (singly and doubly linked lists)
3. Stack and queue implementation using linked lists
4. Binary tree and binary search tree (BST) implementation
5. Tree traversal techniques (in order, preorder, post order)
6. Graph traversal algorithms (DFS and BFS)
7. Implementation of hashing techniques

AI-Integrated Lab Exercises (Indicative)

- GitHub Copilot, ChatGPT / AI assistants, code analysis tools

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 the semester, the final examination is conducted to evaluate a student's performance.

Integrated Professional Core Courses (Theory + Practical)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) every 2 weeks (Minimum 06 assessments including AI-powered Assignment)	20
Mid Semester Examination (MSE)	20
Lab Record	05
Lab Examination	15
Total	**60 marks**

Recommended Tools

- **Programming:** C Programming using VS Code
- **Visualization:** Algorithm Visualization Platforms
- **AI Tools:** GitHub Copilot, ChatGPT / AI assistants, code analysis tools

Suggested Reading

Primary Textbooks

1. Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI, 2014.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis", 4th Edition, Pearson, 2013.

Additional Resources

1. **MIT Open Course Ware - Introduction to Algorithms** https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/

Stanford Algorithms Lectures - Algorithms: Design and Analysis, Part 1 https://online.stanford.edu/courses/soe-ycsalgorithms1-algorithms-design-and-analysis-part-1

CO–PO–PSO Mapping Table

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	3
CO1	3	3	2	2	2	-	-	-	-	1	-	3	2	2
CO2	3	3	3	2	3	-	-	-	-	1	-	3	2	3
CO3	3	3	3	3	3	-	-	-	-	1	-	3	3	3
CO4	3	3	3	3	3	-	-	-	-	1	-	3	3	3
CO5	3	3	3	3	3	1	1	1	1	2	1	3	3	2

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

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112

Department of Computer Science & Medical Engineering

Week-by-Week Teaching Plan (45 Lectures)

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Data Structures, Classification	Lecture + examples	AI-generated classification quiz	Understand what data structures are and their types
	2	C Structures and Union	Lecture + coding demo	AI-assisted code explanation	Explain structure and union memory layout
	3	C Pointers – basics, pointer arithmetic	Lecture + discussion	AI-generated pointer examples	Understand pointer concepts and operations
2	4	Array Definition, Representation, Operations	Lecture	AI visualization support	Explain array storage and basic operations
	5	Insertion, Deletion, Search, Traversal in Arrays	Lecture + demo	AI-assisted code debugging	Implement basic array operations
	6	Two/Multidimensional Arrays, Sparse matrix	Lecture + discussion	AI-generated sparse matrix representation	Understand multi-dimensional arrays and sparse storage
3	7	Stack: Definition, Array Representation	Lecture	AI-based concept mapping	Define stack and its LIFO behavior
	8	Operations on Stacks (push, pop, peek)	Lecture + coding demo	AI-assisted error detection	Implement stack operations in C
	9	Applications: Expression evaluation, Infix to Postfix	Lecture + examples	AI-generated conversion steps	Convert infix to postfix manually
4	10	Infix to Prefix, Recursion	Lecture + discussion	AI-based recursion visualization	Understand recursive algorithms
	11	Tower of Hanoi problem	Case study + demo	AI-supported step-by-step solving	Solve Tower of Hanoi using recursion
	12	Queue: Definition, Representation, Operations	Lecture	AI-generated queue examples	Define queue and its FIFO behavior

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
5	13	Circular Queue	Lecture + demo	AI-assisted circular index calculation	Implement circular queue operations
	14	Applications: Job Scheduling, Maze Problem	Case study + discussion	AI-based scheduling simulation	Apply queues to real-world scheduling
	15	Revision and Problem Solving	Interactive session	AI-generated practice problems	Consolidate Module 1 & 2 concepts
6	16	Linked List: Types, Representation in Memory	Lecture	AI visualization of node links	Understand linked list memory layout
	17	Traversing, Searching in Linked List	Lecture + coding demo	AI-assisted search optimization	Implement traversal and search
	18	Insertion & Deletion in Linked List	Lecture + demo	AI-generated edge case handling	Perform insert/delete operations
7	19	Circular Linked List	Lecture + examples	AI-based circular list visualization	Implement circular linked list
	20	Doubly Linked List, Operations	Lecture + coding demo	AI-assisted pointer management	Implement doubly linked list
	21	Stack & Queue Implementation using Linked Lists	Lecture + discussion	AI-generated code comparison	Compare array vs linked list implementations
8	22	Case Study: Josephus problem	Case study + group discussion	AI-supported simulation	Solve Josephus problem using linked list
	23	Revision and Lab Integration	Lab activity	AI-assisted debugging	Consolidate linked list concepts
	24	Trees: Basic Terminology, Binary Trees	Lecture	AI-generated tree terminology quiz	Define tree terminology
9	25	Binary Tree Representation (Array, Linked)	Lecture + demo	AI visualization support	Represent binary trees in memory

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	26	Complete Binary Trees, Binary Search Trees	Lecture + examples	AI-generated BST examples	Understand BST property
	27	Threaded Binary Trees	Lecture + discussion	AI-based comparison of threaded vs normal	Explain threaded binary trees
10	28	Operations on Binary Trees (Insertion, Deletion)	Lecture + coding demo	AI-assisted BST implementation	Implement BST insert and delete
	29	Search & Traversal (IN order, Preorder, Post order)	Lecture + demo	AI-generated traversal animations	Perform tree traversals
	30	Applications: Expression Evaluation, Game Tree	Case study + discussion	AI-based game tree analysis	Apply trees to expression evaluation
11	31	Greedy Method: Minimum Cost Spanning Trees	Lecture	AI-based MST visualization	Understand MST concepts
	32	Prim's Algorithm	Lecture + coding demo	AI-assisted graph representation	Implement Prim's algorithm
	33	Kruskal's Algorithm	Lecture + demo	AI-generated Union-Find examples	Implement Kruskal's algorithm
12	34	Single Source Shortest Paths: Dijkstra's Algorithm	Lecture + examples	AI-assisted path visualization	Understand Dijkstra's algorithm
	35	Dijkstra's Algorithm Implementation	Lecture + coding demo	AI-generated priority queue integration	Implement Dijkstra's algorithm
	36	Optimal Tree Problem: Huffman Trees	Lecture + discussion	AI-generated compression examples	Understand Huffman coding
13	37	Huffman Coding Algorithm	Lecture + coding demo	AI-assisted tree building	Implement Huffman coding
	38	Complexity Analysis of	Lecture	AI-generated	Compare time/space

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Algorithms		complexity tables	complexity
	39	Choosing the Right Data Structure for a Problem	Problem-solving session	AI-based recommendation system	Select appropriate DS for real problems
14	40	Revision: Arrays, Stacks, Queues	Interactive session	AI-generated practice problems	Consolidate Module 1 & 2
	41	Revision: Linked Lists, Trees	Interactive session	AI-assisted problem solving	Consolidate Module 3 & 4
	42	Revision: Greedy Algorithms and Graph Algorithms	Interactive session	AI-generated exam practice	Consolidate Module 5
15	43	Case Study of Module 1 & 2: Array and ADT Applications	Discussion	AI-supported case analysis	Review array, stack, queue concepts
	44	Case Study of Module 3 & 4: Linked List and Tree Applications	Discussion	AI-assisted data structure analysis	Review linked list and tree concepts
	45	Case Study of Module 5: Graph and Greedy Algorithm Applications	Discussion	AI-based algorithm evaluation	Review MST, Dijkstra, Huffman concepts

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

SUSTAINABLE ENGINEERING & ESG

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2306	**Course Type:** Integrated Program Core Course
Programs Offered: B. Tech. (Common to all CSE Branches)	**Total Hours:** 30 Hours
Semester: III	**Hours/Week:** 02 Hours
Credits: 02	**L – T – P – J:** 2 – 0 –0 – 0
Prerequisites: Fundamental understanding of engineering mathematics, Basic awareness of environmental science	

Course Vision

This course aims to develop a strong foundation in sustainable engineering, Environmental, Social, and Governance (ESG) principles, and digital sustainability systems, enabling students to analyses and address real-world challenges related to energy efficiency, environmental impact, ethical computing, and sustainable digital infrastructure. The course integrates modern analytical tools, AI-based decision-making and data-driven sustainability assessment methods to enhance problem-solving and critical thinking.

Course Objectives

Students will:

1. Understand the fundamental concepts of sustainable engineering, ESG framework, and environmental, social, and governance dimensions of digital and computing systems, including energy usage and environmental impact.
2. Apply sustainability principles, ESG indicators, and basic analytical models to assess environmental performance and resource efficiency in engineering and digital systems.
3. Analyze energy consumption patterns, system efficiency, and environmental impacts in computing infrastructures such as data centers, cloud systems, and software applications.
4. Understand sustainability metrics, lifecycle assessment concepts, and ethical considerations in engineering systems, including responsible use of digital technologies.
5. Evaluate the performance of engineering and digital systems in terms of energy efficiency, sustainability impact, and alignment with ESG goals, using modern analytical and AI-assisted tools.

Course Outcomes (CO)

CO	Course Outcomes	Bloom taxonomy level
CO1	Explain fundamental concepts of sustainability engineering, ESG framework, and analyse environmental impact factors such as energy consumption and resource usage in digital systems.	L3
CO2	Apply sustainability principles, ESG indicators, and analytical models to evaluate environmental performance and efficiency in computing and engineering systems.	L4
CO3	Evaluate energy usage patterns, system efficiency, and resource optimization in IT infrastructure such as cloud systems, data centers, and software applications.	L4
CO4	Analyse sustainability metrics and lifecycle impacts of engineering systems using ESG reporting frameworks, carbon footprint estimation, and responsible design principles.	L3
CO5	Apply sustainability and ESG principles to assess the performance of digital systems in terms of energy efficiency, environmental impact, and ethical governance using AI-assisted tools.	L5

Broad Course Description

Sustainable Engineering and Environmental Governance is an integrated course that introduces the fundamental principles of sustainability science, environmental impact assessment, and ESG (Environmental, Social, and Governance) frameworks to analyses and design responsible digital and engineering systems. The course begins with an understanding of sustainability concepts, ESG principles, and global sustainability goals (SDGs), followed by the evaluation of environmental impacts of computing and engineering systems such as energy consumption, carbon footprint, and resource utilization.

Key topics include green computing, lifecycle assessment (LCA), energy efficiency in IT infrastructure, and sustainable system design, with emphasis on applications in cloud computing, data centers, software systems, and smart infrastructure. The course also explores social and ethical dimensions of technology, including digital inclusivity, data privacy, ethical AI, and algorithmic fairness.

The governance aspect introduces ESG reporting frameworks, sustainability metrics, and compliance standards, enabling students to assess and communicate sustainability performance effectively. AI-based tools are integrated throughout the course for sustainability analytics, carbon estimation, ESG reporting, and energy optimization, enhancing data-driven decision-making.

The course combines theoretical concepts with case studies, AI-assisted analysis, and project-based learning, enabling students to develop analytical skills and responsible engineering judgment required to design and evaluate sustainable digital systems in industries such as IT services, cloud computing, smart cities, and emerging green technologies.

Detailed Course Syllabus

The course consists of five modules. Each module integrates:

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

Course Modules

Module1: Sustainability Fundamentals & ESG (6 Hours)

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.

Module2: 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.

MODULE 3: Social Sustainability & Ethical Computing (6 Hours)

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.

Module 4: Governance, ESG Reporting, Sustainable Design & Future Technologies (6 Hours)

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.

Module 5: Energy Efficiency, Audit & Smart Energy Systems (6 Hours)
Topics: Energy efficiency metrics, Energy losses in computing and electronic systems, Energy-efficient computing: green computing principles, low-power design, Energy Audit: types (preliminary and detailed audit), steps involved, Energy audit of IT infrastructure, 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

AI Integration (AI Pedagogy)
AI Integration:
Module 1 — Sustainability Fundamentals & ESG
Use AI tools for sustainability trend analysis and ESG scoring based on real-world datasets. AI-based visualization of SDGs, ESG indicators, and sustainability performance metrics for organizations.
Module 2 — Environmental Sustainability in Digital Systems
AI-assisted estimation of carbon footprint and energy consumption in IT systems and data centers. Machine learning models for energy prediction, workload optimization, and green cloud computing strategies.
Module 3 — Social Sustainability & Ethical Computing
AI tools for sentiment analysis and stakeholder impact assessment. AI-based detection of bias, fairness issues, and ethical concerns in algorithms and digital systems.
Module 4 — Governance, ESG Reporting, Sustainable Design & Future Technologies
AI-assisted ESG report analysis, KPI tracking, and sustainability dashboards (Power BI/Python tools). AI-based optimization of smart cities, digital twins, Industry 4.0systems, and sustainable IoT infrastructures.
Module 5 — Energy Efficiency, Audit & Smart Energy Systems
AI models for energy consumption prediction and anomaly detection in IT infrastructure. AI-based optimization of data center energy usage, smart energy monitoring using IoT, and identification of energy loss points.

Assessment Scheme
Definitions
1. "Internal Assessment" refers to the Continuous Internal Assessment (CIA). The weightage of this component is 60%. It includes:
   - One Mid Semester Examination (MSE) (Descriptive-based),
   - 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
   - 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.

Professional Core Courses (Theory)

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

***NOTE:**

- If any theory course ***consists of Mini project component***, then the total marks of ***10 (Ten marks)*** shall be evaluated for that mini project and hence there is no APA for such courses.
- If any theory course ***does not consist of mini project***, then APA is mandatory awarded for **10 marks**.

Recommended Tools

- **Simulation & Analysis Tools**
  Energy modeling and system analysis using MATLAB / Python (NumPy, Pandas, Scikit-learn), Lifecycle Assessment (LCA) tools such as OpenLCA / SimaPro
- **Visualization & Dashboard Tools**
  Data visualization using Power BI, Tableau, Python (Matplotlib, Seaborn), ESG dashboards and sustainability reporting tools, Smart city and IoT system visualization platforms
- **AI & Digital Tools**
  AI tools such as ChatGPT for sustainability analysis, report generation, and decision support, Machine learning platforms (TensorFlow, Scikit-learn) for energy prediction and ESG analytics

Suggested Reading

Textbooks

1. Sustainable Computing and the Social Sciences – Lorenz M. Hilty, Bernard Aebischer, 1st Edition, 2015
2. Green IT: Reduce Your Information System's Environmental Impact – Jean-Marc Pierson, 1st Edition, 2012
3. Sustainability Engineering: Concepts, Design and Case Studies – Allen, D. T. & Shonnard, D. R., 1st Edition, 2001.

Reference book

1. Environmental Impact Assessment: Theory and Practice – Glasson, J., Therivel, R., Chadwick, A., 5th Edition, 2012
2. Designing for Sustainability: A Guide to Building Greener Digital Products and Services – Tim Frick, 1st Edition, 2014.

Additional Sources

1. NPTEL – Courses on Sustainable Engineering, Energy Systems, Environmental Studies
2. MIT Open Course Ware – Courses on Sustainability, Energy, and Smart Systems
3. IEEE, Elsevier, and industry ESG reports for latest research and case studies

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**	**3**
CO1	3	2	1	1	1	-	-	-	-	1	-	1	-	-
CO2	2	3	2	2	2	-	-	-	1	1	-	2	-	-
CO3	2	3	2	3	3	-	-	-	1	2	-	2	-	-
CO4	2	3	2	3	2	-	-	-	-	1	-	1	-	-
CO5	2	2	2	2	3	-	-	1	-	1	2	2	-	-
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Department of Computer Science & Medical Engineering

Week by Week Teaching Plan

Week	Lecture No.	Module	Topics Covered	Teaching Method	Expected Learning Outcome
1	1	Module 1	Introduction to Sustainability and Sustainable Development	Lecture + Discussion	Understand sustainability concepts and importance
	2	Module 1	Triple Bottom Line: People, Planet, Profit	Lecture + Case Study	Explain the Triple Bottom Line framework
	3	Module 1	ESG Framework and Components	Lecture + PPT	Describe ESG principles and components
2	4	Module 1	Sustainable Development Goals (SDGs)	Seminar + Discussion	Understand global sustainability goals
	5	Module 1	ESG vs CSR Concepts	Comparative Analysis	Differentiate ESG and CSR practices
	6	Module 1	Sustainability Challenges in Digital Economy	Lecture + Case Analysis	Identify sustainability issues in IT systems
3	7	Module 1	Green Computing and Sustainable IT Infrastructure	Lecture + Demonstration	Understand green computing practices
	8	Module 1	Energy-Efficient Data Centers	Lecture + Video Demo	Analyze energy-efficient infrastructure
	9	Module 1	Case Studies on Sustainable Engineering Applications	Case Study	Apply sustainability concepts in engineering
4	10	Module 2	Environmental Sustainability in Digital Systems	Lecture	Understand environmental impacts of digital systems
	11	Module 2	Carbon Footprint of IT Systems	Lecture + Examples	Analyze carbon emissions in IT
	12	Module 2	Energy Consumption in Data Centers	Lecture + Numerical Examples	Evaluate energy usage in data centers
5	13	Module 2	Lifecycle Assessment (LCA) Concepts	Lecture + Activity	Understand lifecycle assessment methods
	14	Module 2	E-Waste Management Techniques	Lecture + Discussion	Explain e-waste management strategies
	15	Module 2	Green Cloud Computing	Lecture + Case Study	Understand sustainable cloud technologies
6	16	Module 2	Energy-Efficient Servers	Demonstration + Lecture	Analyze efficient server technologies
	17	Module 2	Cloud Optimization Techniques	Lecture + Practical Examples	Optimize cloud resource utilization
	18	Module 2	Sustainable Software Engineering	Lecture + Discussion	Apply sustainability principles in software design
7	19	Module 3	Introduction to Social Sustainability	Lecture	Understand social sustainability principles

	20	Module 3	Stakeholder Analysis	Lecture + Activity	Identify stakeholders in digital systems
	21	Module 3	Digital Inclusivity and Accessibility	Lecture + Demonstration	Explain inclusive digital system design
8	22	Module 3	Data Privacy and Protection	Lecture + Case Study	Understand privacy issues in computing
	23	Module 3	Ethical AI Concepts	Lecture + Discussion	Explain ethics in AI systems
	24	Module 3	Cybersecurity and Society	Lecture + Examples	Analyze cybersecurity social impacts
9	25	Module 3	Social Impact of Automation	Seminar + Discussion	Evaluate automation effects on society
	26	Module 3	Fairness in Algorithms	Lecture + Case Study	Understand algorithmic fairness
	27	Module 3	Inclusive Software Design	Practical Demonstration	Design accessible and inclusive applications
10	28	Module 4	Corporate Governance Fundamentals	Lecture	Understand governance principles
	29	Module 4	ESG Reporting Frameworks: GRI and TCFD	Lecture + Examples	Explain ESG reporting standards
11	31	Module 4	Sustainability Data Visualization	Demonstration + Tools	Visualize sustainability metrics
	32	Module 4	Circular Economy Concepts	Lecture + Discussion	Understand circular economy models
	33	Module 4	Sustainable Software Design	Lecture + Examples	Apply sustainable software principles
12	34	Module 4	Green AI and Energy-Efficient AI Systems	Lecture + Case Study	Analyze energy-efficient AI models
	35	Module 4	Smart Cities and Industry 4.0 Sustainability	Lecture + Video Demo	Understand smart sustainable systems
	36	Module 4	Digital Twins for Sustainability	Lecture + Demonstration	Explain digital twin applications
13	37	Module 5	Introduction to Energy Efficiency Metrics	Lecture	Understand energy efficiency measurements
	38	Module 5	Energy Losses in Computing Systems	Lecture + Numerical Examples	Analyze power losses in computing
	39	Module 5	Green Computing Principles and Low-Power Design	Lecture + Discussion	Apply low-power computing concepts
14	40	Module 5	Energy-Efficient Computing Systems	Lecture	

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

OOPS WITH JAVA

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

SEMESTER – III

Subject Code	:	25SM2308	**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P-J	:	1–1–0-0			

Course Learning Objectives:

This Course will enable students to:

1. Introduce the basics of data mining, including statistical modelling, machine learning, and computational approaches.
2. Learn to extract relevant features from data for improved model performance.
3. Study the stream data model and techniques for sampling, filtering, and counting in data streams.
4. Understand market-basket analysis and algorithms for discovering frequent item sets.
5. Understand eigenvalues, eigenvectors, PCA, and SVD for reducing data dimensionality.

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** means it includes not only traditional lecture method, but different type 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 analyse information rather than simply recall it.
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	**06 Hours**

Basics of Java:

Introduction to Java -Overview of Java's history, features, and architecture.
(Text Book -1 Chapter 1)

Basics of Java Programming-Fundamentals of Java syntax, datatypes, and operators.
(TextBook-1Chapter2,3,4)

Control Statements -Using conditional and looping statements to control program flow.
(TextBook-1Chapter5)

UNIT – II	**06 Hours**

OOP Concepts:

Classes and Objects-Defining classes and creating objects in Java.
(TextBook-1 Chapter 6)

Methods and Constructors -Creating and using methods and constructors.
(Text Book -1 Chapter 7)

InheritanceandPolymorphism-Implementinginheritanceandpolymorphic behavior.
(TextBook-1Chapter8)

Abstract Classes and Interfaces -Using abstract classes and interfaces to define templates.

(TextBook-1Chapter9)	
UNIT – III	**06 Hours**
Exceptions, Data Structures and Collections Framework **Exception Handling -** Handling errors using Java's exception handling mechanisms. **(Text Book -1 Chapter 10)** **Arrays and Strings -** Working with arrays and string manipulation. **(Text Book -1 Chapter 3,16)** **Generics -** Implementing generic types to enhance code reusability. **(Text Book -1 Chapter 14)**	
UNIT – IV	**06 Hours**
Multithreading and Concurrency **Multithreading Concepts and Implementation -** Creating and managing multiple threads. **(Text Book -1 Chapter 11)** **Synchronization -** Synchronizing threads to avoid concurrency issues. **(Text Book -1 Chapter 11)** **Concurrency Utilities -** Utilizing Java's concurrency utilities for better thread management. **(Text Book -1 Chapter 15)**	
UNIT – V	**06 Hours**
Java I/O and New Features **File I/O -** Reading from and writing to files in Java. **(Text Book -1 Chapter 20)** **JDBC –** Introducing JDBC, Getting Started. **(Text Book-2 Chapter 1 and 2)**	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	**Develop** simple java programs that make use of classes and objects	L3
CO2	**Write** Java application programs using OOP principles and proper program structuring	L3
CO3	**Make** use of inheritance and interfaces to develop java application	L3
CO4	**Model** exception handling, multi-threading concepts in java.	L3
CO5	**Develop** Java program that connects to a database and be able to perform various operations.	L3

Text books:

1. Herbert Schildt, The Complete Reference, Java, 9th Edition, Oracle Press,2014
2. Tulika Das, Oracle Database JDBC Developers Gude, 23c, Oracle, 2024

Web links and Video Lectures (e-Resources):

1. https://docs.oracle.com/javase/tutorial/java/index.html
2. https://docs.oracle.com/cd/E19957-01/817-2177/djjdbc.html

Activity Based Learning (Suggested Activities in Class)

1. Demonstration of solution to a problem through programming.
2. Providing assignments and asking to discuss as seminar presentations.

Lab Experiments for Object-Oriented Programming with Java

1. Basic Java Program Write, compile, and run a simple Java program that prints "Hello, World!". Experiment with different data types and basic operations.
2. Control Structures Implement a Java program using various control structures (if-else, switch, loops). Create a program to calculate the factorial of a number using different loop types.
3. Arrays and Strings Write a Java program to manipulate arrays (sorting, searching, merging). Implement string operations such as concatenation, substring extraction, and string reversal.
4. Classes and Objects Create a class representing a real-world object with attributes and methods. Instantiate objects and demonstrate the use of constructors and destructors.
5. Static Members and Methods Write a Java program to demonstrate the use of static variables and methods. Implement a singleton class in Java.
6. Inheritance and Polymorphism Implement a Java program demonstrating single and multilevel inheritance. Demonstrate method overriding and overloading. Create a program using abstract classes and interfaces.
7. Exception Handling Write a Java program that demonstrates the use of try-catch blocks. Implement custom exception handling.
8. File Handling Write a Java program to read from and write to a file. Implement a program that handles file exceptions.
9. Collections Framework Create a program that uses Array List, LinkedList, HashSet, and HashMap. Perform operations like adding, removing, and iterating through elements.
10. Mini Project Develop a mini-project incorporating the concepts learned throughout the course.

AI Integration

Module–1: Basics of Java

AI enhances understanding of Java fundamentals including syntax, data types, operators, and control statements through intelligent code generation, syntax correction, automated debugging, and interactive coding assistants. Students use AI-supported programming tools to visualize program execution and improve logical reasoning.

Module–2: OOP Concepts

AI supports learning object-oriented programming concepts such as classes, objects, inheritance, polymorphism, abstract classes, and interfaces through AI-assisted UML generation, code refactoring, intelligent class design suggestions, and automated code reviews.

Module–3: Exceptions, Data Structures and Collections Framework

AI facilitates exception analysis, debugging, and optimization of arrays, strings, generics, and collections. Intelligent assistants help students understand runtime errors, improve code efficiency, and apply reusable programming structures.

Module–4: Multithreading and Concurrency

AI assists students in understanding thread management, synchronization, and concurrency utilities through visual thread simulations, deadlock detection support, performance monitoring, and AI-driven concurrency debugging tools.

Module–5: Java I/O and JDBC

AI supports implementation of file handling and database connectivity using intelligent query generation, automated JDBC code assistance, database debugging support, and AI-powered data handling workflows.

Authentic Intelligence Reflection

Module–1: Basics of Java

Encourages students to critically analyze programming logic, evaluate syntax correctness, and apply structured problem-solving techniques while developing Java programs.

Module–2: OOP Concepts

Promotes analytical thinking in designing reusable and scalable object-oriented systems using abstraction, inheritance, and polymorphism for real-world software applications.

Module–3: Exceptions, Data Structures and Collections Framework

Develops authentic understanding of error handling, generic programming, and efficient data management for building robust and maintainable software systems.

Module–4: Multithreading and Concurrency

Encourages thoughtful analysis of concurrent execution, synchronization challenges, and thread coordination in performance-critical applications.

Module–5: Java I/O and JDBC

Helps students understand practical implementation of file systems and database connectivity while emphasizing secure data handling and efficient information management.

AI-Powered Assignments

Assignment 1: AI-Generated Java Syntax and Control Flow Analyzer

Prompt for Students

"Generate Java code for a console-based application that demonstrates Java syntax, operators, loops, and conditional statements for a student grading system."

Student Tasks

1. Analyze whether the generated code correctly implements operators, conditions, and loops.
2. Identify syntax errors, redundant logic, and inefficient control structures.
3. Improve the program using modular methods, optimized conditions, and proper input validation.

Assignment 2: AI-Assisted OOP Design System

Prompt for Students

"Generate Java code using object-oriented programming concepts to create a library management system implementing inheritance, polymorphism, abstract classes, and interfaces."

Student Tasks

1. Verify whether classes, objects, and inheritance are implemented correctly.
2. Identify issues such as poor encapsulation, improper abstraction, and code duplication.
3. Improve the design using interfaces, method overriding, constructor optimization, and reusable components.

Assignment 3: AI-Based Exception Handling and Collections Framework

Prompt for Students

"Generate Java code demonstrating exception handling, arrays, strings, generics, and collections framework for a student record management system."

Student Tasks

1. Analyze whether exception handling mechanisms are correctly implemented.
2. Identify limitations such as unchecked exceptions, improper collection usage, and inefficient searching.
3. Improve the code using generic collections, custom exceptions, and optimized data handling.

Mini Project Components

Project 1: Student Information Management System

Problem Statement

Develop a Java-based student information management system using object-oriented programming concepts, collections framework, and exception handling.

Features

- Student registration and record management
- Search and update operations
- Exception handling for invalid inputs
- File storage support

Assessment Focus

- Understanding Java fundamentals
- OOP implementation
- Error handling and collections usage

Project 2: Banking Transaction Management System

Problem Statement

Develop a multithreaded banking application that supports concurrent transactions using synchronization and thread management.

Features

- Account creation and balance tracking
- Deposit and withdrawal operations
- Concurrent transaction handling
- Thread synchronization

Assessment Focus

- Multithreading concepts
- Synchronization techniques
- Concurrency utilities

Project 3: Library Management System with JDBC

Problem Statement

Build a Java application integrated with MySQL database for library management operations.

Features

- Book issue and return management
- JDBC connectivity
- Database CRUD operations
- User authentication

Assessment Focus

- JDBC implementation
- Database interaction
- File and data handling

CO, PO and PSO mapping Table:

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**	**3**
CO1	3	2	-	-	2	-	-	1	-	-	-	2	1	1
CO2	3	3	2	1	3	-	-	1	1	1	-	3	2	2
CO3	3	3	3	2	3	-	-	1	1	1	-	3	2	2
CO4	3	3	3	3	3	1	-	2	2	1	1	3	2	3
CO5	3	3	3	3	3	1	1	2	2	2	2	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

45 Hours Weekly Lecture Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Java and History	Lecture + Discussion	AI-generated Java timeline	Understand Java evolution
	2	Java Features and Architecture	Lecture + PPT	AI-assisted architecture diagrams	Explain Java architecture
	3	Java Program Structure	Coding Demo	AI code suggestions	Write basic Java programs
2	4	Data Types and Variables	Lecture + Examples	AI-generated examples	Use Java data types
	5	Operators in Java	Problem Solving	AI-assisted operator evaluation	Apply operators
	6	Conditional Statements	Coding Demo	AI-generated conditions	Implement decision making
3	7	Looping Statements	Lecture + Coding	AI-generated loop tracing	Develop iterative programs
	8	Nested Loops	Problem Solving	AI-assisted debugging	Solve nested loop problems
	9	Control Transfer Statements	Lecture	AI-generated flowcharts	Use break and continue
4	10	Classes and Objects	Lecture + Demo	AI-based class generation	Create classes and objects
	11	Constructors	Coding Demo	AI-generated constructor examples	Implement constructors
	12	Methods in Java	Problem Solving	AI-assisted method suggestions	Develop reusable methods
5	13	Inheritance	Lecture +	AI-generated hierarchy	Apply inheritance

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
			Diagram	diagrams	
	14	Polymorphism	Coding Demo	AI-assisted polymorphism examples	Implement polymorphism
	15	Method Overloading & Overriding	Problem Solving	AI-generated comparisons	Differentiate overload and override
6	16	Abstract Classes	Lecture	AI-generated templates	Design abstract classes
	17	Interfaces	Coding Demo	AI-generated interface examples	Implement interfaces
	18	Packages in Java	Lecture + Demo	AI-assisted package management	Organize Java programs
7	19	Exception Handling Basics	Lecture	AI-generated exception cases	Handle exceptions
	20	Try-Catch-Finally	Coding Demo	AI-assisted debugging	Implement exception handling
	21	User-defined Exceptions	Problem Solving	AI-generated examples	Create custom exceptions
8	22	Arrays in Java	Lecture + Coding	AI-generated array operations	Use arrays effectively
	23	Strings and String Methods	Coding Demo	AI-assisted string manipulation	Process strings
	24	Generics	Lecture + Examples	AI-generated generic classes	Develop reusable code
9	25	Introduction to Collections	Lecture	AI-generated collection diagrams	Understand collections
	26	List and Set Interfaces	Coding Demo	AI-assisted collections usage	Use lists and sets
	27	Map Interface	Problem Solving	AI-generated hash map examples	Implement maps
10	28	Introduction to Threads	Lecture + Demo	AI-generated thread lifecycle	Understand multithreading
	29	Creating Threads	Coding Demo	AI-assisted thread creation	Develop thread programs
	30	Thread Lifecycle	Lecture	AI-generated visualization	Analyze thread states
11	31	Synchronization	Lecture + Coding	AI-assisted synchronization examples	Avoid race conditions
	32	Inter-thread	Coding Demo	AI-generated	Coordinate threads

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Communication		communication examples	
	33	Concurrency Utilities	Lecture	AI-assisted concurrent utilities	Use Java concurrency APIs
12	34	File Handling Basics	Lecture + Demo	AI-generated file examples	Perform file operations
	35	File Reading and Writing	Coding Demo	AI-assisted I/O handling	Implement file processing
	36	Serialization	Lecture + Examples	AI-generated serialization demo	Store object data
13	37	JDBC Introduction	Lecture	AI-generated DB connectivity flow	Understand JDBC
	38	Database Connectivity	Coding Demo	AI-assisted SQL generation	Connect Java to databases
	39	CRUD Operations using JDBC	Coding Demo	AI-generated SQL queries	Perform database operations
14	40	Advanced JDBC Concepts	Lecture + Demo	AI-assisted prepared statements	Secure database applications
	41	Mini Project Development	Project-based Learning	AI-assisted coding support	Build integrated applications
	42	Testing and Debugging	Practical Session	AI-based debugging tools	Debug Java programs
15	43	Case Study Discussion	Discussion	AI-generated case analysis	Analyze real-world applications
	44	Course Revision	Interactive Session	AI-generated quizzes	Consolidate Java concepts
	45	Final Problem Solving Session	Q&A + Coding	AI-assisted practice problems	Apply complete course knowledge

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

IOT APP DEVELOPMENT FOR MEDICAL WEARABLES

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

SEMESTER – III

Subject Code	:	25SM2309	**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P-J	:	1–1–0-0			

Course Learning Objectives:

This Course will enable students to:

6. Basic fund a mental concept of Analog & Digital Circuits, Microprocessor&MicrocontrollerwiththeintroductionofEmbeddedSystems, Understand the fundamentals of the Internet of Things and its significance
7. Demonstration of using Analog & digital Sensors & actuators along with Arduino for data Acquisition
8. Learn the basics of using Tinker cad for IoT simulation.
9. Explore various IoT components, sensors, actuators, and communication protocols.

Teaching-Learning Process (General Instructions):

Thesearesamplenewpedagogicalmethodsthatteacherscanusetoacceleratethe attainment of the various course outcomes.

1. **Real-World Examples:** Use real-**world examples** and scenarios to demonstrate the practical relevance of Java programming concepts which enhances their understanding of how Java is used in real-world applications
2. **Interactive Coding Sessions:** Conduct **Interactive coding sessions** where students can code alongside the teacher or participate in coding challenges. This promotes active participation and helps students develop their coding skills
3. **Project-Based Learning**: Assign projects or mini-projects that require students to apply Java programming concepts and develop complete applications. This approach fosters independent learning, problem solving skills and a deeper understanding of Java Programming Principles.
4. **Active Learning Strategies:** Incorporate **active learning strategies** such as group discussions, problem-solving activities, case studies, and hands-on coding exercises. This allows students to actively engage with the material and apply their knowledge in practical scenarios
5. **Collaborative Learning:** Encourage students to work in pairs or small groups on programming tasks. **Collaborative learning** promotes peer-to-peer learning, fosters teamwork, and allows for the exchange of ideas and knowledge.

UNIT – I	**06 Hours**

Introduction to IoT and Tinker cad:

Basics of Analog & Digital Circuits using Tinker cad. Brief History of Microprocessor, Microcontroller, Embedded Systems, Definition and importance of IoT, IoT Components and communication: Sensors and actuators, Communication protocols (WIFI, Bluetooth, MQTT), Trends in the Adoption of IoT in modern applicants, Risks, Privacy, and Security. Sensor Networks, Sensors and actuators, Analog/Digital Conversion

(Textbook-1:Ch2,Ch3).

UNIT – II	**06 Hours**

Fundamentals of Arduino & raspberry Pi: Architecture of Arduino and raspberry Pi3, basic concepts of Embedded C Programming & Arduino

Department of Computer Science & Medical Engineering

UNIT – III	06 Hours
IoT Data Acquisition: Analog and digital sensors, reading sensor data using Tinker cad Using Tinker cad software for designing & debugging of Analog, Digital & mixed Circuits ***(Text book 1: Ch 4, Ch 5).***	
UNIT – IV	**06 Hours**
IoT Project Development I: Designing an IoT project concept, Setting up the simulation environment ***IoT Project Development II:*** Implementing the IoT project using Tinker cad, temperature &humiditymeasurementusingRaspberryPi3&DHT11Sensorusing Python Programming.	
UNIT – V	**06 Hours**
IoT Applications and Case Studies on Medical wearables: Smart home systems, Industrial IoT, Healthcare applications. ***Final Project and Presentation:*** *Developing an advanced IoT project, Presenting the project and its functionalities*	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	**Identification** of different Analog & Digital circuits, study on the various controllers	L2
CO2	**Utilize** the Features & Architecture of Arduino & raspberry Pi3 for reading data from sensors	L3
CO3	**Analyse** the working of Software tool to design analog, digital & mixed circuits	L4
CO4	**Examine** problem-solving and critical thinking skills for IoT applications	L4
CO5	**Design** Smart circuits in real time environment using Arduino & RaspberryPi	L6

Text books:

1. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things "by David Hanes, et al.

Web links and Video Lectures (e-Resources):

1. Online tutorials and documentation on Tinkercad and Arduino.
2. *Tinkercad:www.tinkercad.com/circuits*

Activity Based Learning (Suggested Activities in Class)

1. Using Hardware components Lab exercises and assignments to assess practical skills
2. Mid-term project to gauge project development abilities
3. Class participation and discussions on IoT concept s and trends

List of experiments (Simulation and/or Hardware):

1. Building basic electronic circuit using Tinker cad software.
2. ControllingblinkingofLEDusing555 Timer
3. VerificationofallBooleanexpressionusingbasicgates&universalGates.
4. Using flip flops for various counters
5. Stoptimerusing555&digitalIC's

List of Projects using Tinker cad-IoT
(For Both Simulation & Hardware using Arduino & RaspberryPi3)

1. Smart Parking system.
2. Smart Notice Board for announcement in classroom
3. Sensing temperature displaying the same on LCD.
4. 4-waytrafficlightdesigning&automating.
5. Sun tracking solar Panel.
6. Generating National anthem of Bharath using Arduino

CO–PO Mapping Table

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**	**3**
CO1	3	2	1	1	2	2	1	2	1	1	1	2	2	1
CO2	3	2	3	2	3	1	1	1	2	1	1	3	2	3
CO3	3	3	3	2	3	1	1	1	2	2	1	3	3	3
CO4	3	3	3	3	3	2	1	1	2	2	2	3	3	3
CO5	2	3	3	2	3	3	2	2	3	3	2	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Week by week Teaching _ Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to IoT and Embedded Systems	Lecture + discussion	AI-generated IoT application examples	Understand the concept and importance of IoT systems
	2	History of Microprocessors and Microcontrollers	Lecture + examples	AI-assisted timeline visualization	Explain the evolution of embedded systems
	3	IoT Components and Communication Basics	Lecture + demo	AI-generated IoT architecture diagrams	Identify IoT components and communication methods
2	4	Sensors and Actuators in IoT	Lecture + hardware demo	AI-assisted sensor classification	Understand different sensors and actuators used in IoT
	5	Communication Protocols: WiFi,	Lecture + case study	AI-generated protocol	Differentiate IoT communication

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Bluetooth, MQTT		comparison	protocols
	6	Risks, Privacy, and Security in IoT	Lecture + discussion	AI-assisted cybersecurity examples	Understand IoT privacy and security challenges
3	7	Basics of Analog and Digital Circuits using Tinker cad	Lecture + simulation	AI-generated circuit visualization	Build basic analog and digital circuits
	8	Analog to Digital and Digital to Analog Conversion	Lecture + simulation	AI-assisted ADC/DAC examples	Explain analog and digital conversion concepts
	9	Introduction to Sensor Networks	Lecture + examples	AI-generated sensor network models	Understand sensor network architecture
4	10	Introduction to Arduino Architecture	Lecture + demo	AI-assisted Arduino board visualization	Explain Arduino architecture and components
	11	Basics of Embedded C Programming	Lecture + coding practice	AI-generated code suggestions	Write simple Embedded C programs
	12	Arduino IDE Setup and Programming	Lecture + hands-on	AI-assisted debugging support	Develop and execute Arduino programs
5	13	Raspberry Pi 3 Architecture	Lecture + examples	AI-generated Raspberry Pi diagrams	Understand Raspberry Pi architecture
	14	Introduction to Python Programming for IoT	Lecture + coding practice	AI-assisted Python code generation	Develop basic Python programs for IoT
	15	Interfacing Components with Raspberry Pi	Lecture + demo	AI-assisted interfacing examples	Interface sensors and peripherals with Raspberry Pi
6	16	IoT Data Acquisition Concepts	Lecture + examples	AI-generated data flow diagrams	Understand IoT data acquisition process
	17	Analog Sensors in IoT	Lecture + simulation	AI-assisted sensor calibration	Read and process analog sensor data
	18	Digital Sensors in IoT	Lecture + simulation	AI-generated sensor output analysis	Read and interpret digital sensor data
7	19	Reading Sensor Data using Tinker cad	Hands-on lab	AI-assisted simulation	Acquire sensor data using Tinker cad

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
				debugging	
	20	Designing Analog Circuits in Tinker cad	Simulation + practice	AI-generated circuit suggestions	Design analog circuits using Tinker cad
	21	Designing Digital Circuits in Tinker cad	Simulation + practice	AI-assisted logic analysis	Build digital circuits using Tinker cad
8	22	Mixed Circuit Design in Tinker cad	Lecture + simulation	AI-generated mixed-circuit examples	Create mixed analog-digital circuits
	23	Debugging IoT Circuits in Tinker cad	Hands-on practice	AI-assisted troubleshooting	Debug IoT circuits effectively
	24	Introduction to IoT Project Development	Lecture + discussion	AI-generated project ideas	Understand IoT project development stages
9	25	Designing an IoT Project Concept	Group activity	AI-assisted brainstorming	Design an IoT-based solution for real-world problems
	26	IoT Project Planning and Workflow	Lecture + case study	AI-generated workflow models	Plan IoT project implementation
	27	Setting up Simulation Environment in Tinker cad	Hands-on practice	AI-assisted setup guidance	Configure simulation environment for IoT projects
10	28	Temperature Measurement using DHT11 Sensor	Lecture + demo	AI-generated sensor data visualization	Interface DHT11 sensor for temperature monitoring
	29	Humidity Measurement using DHT11 Sensor	Lecture + coding practice	AI-assisted humidity analysis	Measure humidity using DHT11 sensor
	30	Raspberry Pi and DHT11 Interfacing using Python	Hands-on coding	AI-assisted Python debugging	Develop Python programs for sensor interfacing
11	31	IoT Project Implementation using Tinker cad	Project-based learning	AI-assisted implementation suggestions	Implement IoT projects using Tinker cad
	32	Data Monitoring and Visualization	Lecture + demo	AI-generated dashboards	Visualize IoT sensor data effectively
	33	Troubleshooting IoT Systems	Lecture + discussion	AI-assisted fault diagnosis	Identify and resolve IoT system issues

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
12	34	Smart Home IoT Applications	Lecture + case study	AI-generated smart home simulations	Understand IoT applications in smart homes
	35	Industrial IoT Applications	Lecture + examples	AI-assisted industrial analytics	Explain industrial IoT applications
	36	Healthcare IoT Applications	Lecture + case study	AI-generated wearable device analysis	Understand IoT applications in healthcare
13	37	Medical Wearables and IoT	Lecture + discussion	AI-assisted health monitoring examples	Explain the role of IoT in wearable healthcare devices
	38	Case Study: Smart Healthcare Monitoring System	Case study	AI-generated patient monitoring simulation	Analyze IoT healthcare monitoring systems
	39	Case Study: Industrial Automation using IoT	Case study	AI-assisted automation workflow	Understand IoT-enabled industrial automation
14	40	Advanced IoT Project Development	Project work	AI-assisted optimization suggestions	Build advanced IoT solutions
	41	IoT Project Integration and Testing	Hands-on practice	AI-assisted testing support	Test and integrate IoT system modules
	42	Project Documentation and Reporting	Lecture + activity	AI-generated report templates	Prepare professional IoT project documentation
15	43	Final IoT Project Demonstration	Project presentation	AI-assisted performance analysis	Demonstrate working IoT projects
	44	Project Evaluation and Feedback	Discussion + review	AI-generated evaluation rubrics	Evaluate IoT project effectiveness
	45	Course Review and Future Trends in IoT	Q&A + discussion	AI-generated future IoT trends	Summarize IoT concepts and explore future applications

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DIFFERENTIAL EQUATIONS AND ENGINEERING OPTIMIZATION

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2401	**Course Type:** Mandatory Core Course
Programs Offered: CSME	**Total Hours:** 45 Hours
Semester: IV	**Hours/Week:** 03 Hours
Credits: 03	**L – T – P – J:** 3 – 0 –0 – 0
Prerequisites: Linear Algebra & Data Foundations, Calculus & Mathematical Modelling, Applied Statistics & Stochastic Processes	

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.

Course Description

This course provides a comprehensive foundation in advanced mathematical techniques and optimization methods essential for engineering, science, and computational applications. The course begins with vector calculus concepts including gradient, divergence, curl, vector fields, and integral theorems, enabling students to model and analyze multidimensional physical systems. It then introduces numerical methods for solving ordinary and partial differential equations, focusing on computational techniques such as Euler, Runge-Kutta, multistep, and boundary value problem methods used in scientific and engineering simulations.

Course Outcomes (CO)

At the end of course, students will be able to:

CO	Course Outcomes	Bloom taxonomy level
CO1	Apply vector calculus concepts such as gradient, divergence, curl, and vector integrals to solve engineering and computational problems.	L3
CO2	Implement numerical methods for solving ordinary and partial differential equations arising in engineering and scientific models.	L3
CO3	Formulate and solve classical optimization problems including single and multivariable optimization with constraints.	L4
CO4	Develop and solve linear programming models for real-world problems	L3
CO5	Apply nonlinear optimization techniques such as gradient descent, Fletcher–Reeves method, and penalty methods to engineering applications.	L5

Teaching Learning Process

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.

Course Modules

Module 1: Vector Calculus **[09 Hours]**

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 2: Numerical Methods for ODE and PDE **[09 Hours]**

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 3: Introduction to Optimization (Classical Optimization) **[09 Hours]**

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 4: Linear Programming **[09 Hours]**

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 5: Nonlinear Optimization **[09 Hours]**

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

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)					
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)					

S. No	Exercise Title	Aim / Objective	Tools	Hrs	CO
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

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration**
  - Gradient calculations in machine learning optimization
  - Simulation of dynamic systems in robotics
  - Decision optimization in intelligent systems
  - Linear programming used in decision-making systems
  - Gradient descent optimization algorithms for training neural networks and deep learning
- **Authentic Intelligence Reflection**
  - Understanding limitations of mathematical approximations in AI systems
  - Evaluating numerical errors in computational models
  - Ethical considerations in automated optimization systems
  - Ethical decision-making in optimization problems
  - Understanding biases introduced by optimization algorithms

RECOMMENDED TOOLS

1. Python
2. MATLAB
3. NumPy
4. Pandas
5. Matplotlib
6. Scikit-learn

SUGGESTED READING

TEXT BOOKS:

1. Hamdy A. Taha, Operations Research: An Introduction, 2017, 10th Edition, Pearson Education, Inc.
2. Rao, S.S., Engineering optimization: theory and practice, 2019, 5th 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.

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	3
CO1	3	2	1	–	1	–	–	–	–	–	–	3	1	1
CO2	3	3	2	2	2	–	–	–	–	–	–	3	2	2
CO3	3	3	3	2	2	1	–	–	1	–	1	3	2	1
CO4	3	3	3	3	3	1	–	–	1	1	1	3	3	2
CO5	3	3	3	3	3	1	1	1	1	1	1	3	3	3

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

Week by Week Teaching Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Vector Calculus	Lecture + Discussion	AI-generated vector field visualization	Understand fundamentals of vector calculus
	2	Scalar and Vector Valued Functions	Lecture + Examples	AI-assisted graph plotting	Differentiate scalar and vector functions
	3	Gradient and Tangent Plane	Problem Solving	AI-generated surface visualization	Compute gradients and tangent planes
2	4	Directional Derivative	Lecture + Numerical Examples	AI-assisted derivative computation	Apply directional derivatives
	5	Divergence and Curl	Lecture + Visualization	AI-generated fluid flow models	Analyze divergence and curl
	6	Scalar and Vector Potentials	Lecture + Discussion	AI-assisted potential field visualization	Understand scalar/vector potentials
3	7	Vector Identities	Problem Solving	AI-generated symbolic simplification	Apply vector identities
	8	Line Integrals	Lecture + Examples	AI-assisted integration steps	Evaluate line integrals
	9	Surface and Volume Integrals	Lecture + Practice	AI-generated 3D integral visualization	Compute surface and volume integrals
4	10	Green's Theorem	Lecture + Applications	AI-generated theorem demonstrations	Apply Green's theorem

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	11	Stokes' Theorem	Lecture + Examples	AI-assisted vector surface visualization	Solve problems using Stokes' theorem
	12	Gauss Divergence Theorem	Problem Solving	AI-generated divergence simulations	Apply Gauss divergence theorem
5	13	Introduction to ODEs and PDEs	Lecture + Discussion	AI-generated differential equation plots	Understand ODE/PDE concepts
5	14	Existence and Uniqueness of Solutions	Lecture	AI-assisted symbolic verification	Analyze solution existence
	15	Taylor Series Method	Problem Solving	AI-generated Taylor approximations	Solve ODEs using Taylor series
6	16	Euler's Method	Coding Demo	AI-assisted numerical computation	Implement Euler's method
	17	Modified Euler Method	Lecture + Practice	AI-generated iterative calculations	Solve ODEs using modified Euler
	18	Runge-Kutta Methods	Lecture + Examples	AI-assisted RK visualization	Apply Runge-Kutta methods
7	19	Multistep Methods	Lecture + Numerical Problems	AI-generated solution comparison	Understand multistep approaches
	20	Higher Order Equations and Systems	Lecture + Examples	AI-assisted system solving	Solve higher-order systems
	21	Stability Analysis	Lecture + Discussion	AI-generated stability plots	Analyze numerical stability
8	22	Boundary Value Problems	Lecture + Examples	AI-assisted BVP solving	Understand boundary value problems
	23	Shooting Method	Problem Solving	AI-generated iterative visualization	Apply shooting method
	24	Difference and Variational Methods	Lecture + Practice	AI-assisted discretization	Solve BVPs numerically
9	25	Introduction to Linear Programming	Lecture + Discussion	AI-generated optimization models	Understand LP concepts
9	26	Standard Form of LP	Lecture + Examples	AI-assisted LP formulation	Convert problems into standard form
	27	Geometry of LP Problems	Visualization + Lecture	AI-generated feasible regions	Analyze geometric interpretation

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
10	28	Graphical Method	Problem Solving	AI-assisted graph plotting	Solve LP problems graphically
	29	Simplex Algorithm	Lecture + Demo	AI-generated simplex tables	Apply simplex method
	30	Revised Simplex Method	Lecture + Practice	AI-assisted optimization steps	Understand revised simplex
11	31	Duality in Linear Programming	Lecture + Discussion	AI-generated primal-dual comparison	Analyze LP duality
	32	Resource Allocation Applications	Case Study	AI-assisted scheduling optimization	Apply LP in real-world systems
	33	Transportation and Logistics Optimization	Problem Solving	AI-generated transportation models	Solve logistics optimization problems
12	34	Introduction to Nonlinear Optimization	Lecture + Discussion	AI-generated nonlinear surfaces	Understand nonlinear optimization
	35	Univariate Optimization Methods	Lecture + Examples	AI-assisted minima/maxima detection	Apply univariate methods
	36	Gradient of a Function	Lecture + Practice	AI-generated gradient visualization	Compute gradients for optimization
13	37	Gradient Descent Method	Coding Demo	AI-assisted convergence visualization	Implement gradient descent
	38	Fletcher-Reeves Method	Lecture + Numerical Examples	AI-assisted optimization tracing	Apply conjugate gradient methods
	39	Constrained Optimization Problems	Lecture + Discussion	AI-generated feasible constraint plots	Analyze constrained optimization
14	40	Cutting Plane Method	Lecture + Problem Solving	AI-assisted constraint reduction	Apply cutting plane techniques
	41	Interior Penalty Function Method	Lecture + Examples	AI-generated penalty visualization	Solve constrained problems
	42	Exterior Penalty Function Method	Problem Solving	AI-assisted optimization support	Apply exterior penalty methods
15	43	Optimization in Neural Networks	Case Study	AI-assisted ML optimization demo	Relate optimization to AI training
	44	Course Revision and	Interactive	AI-generated quizzes	Consolidate course

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Applications	Session	and summaries	concepts
	45	Final Problem Solving & Discussion	Q&A + Practice	AI-assisted solution evaluation	Apply complete optimization and numerical methods knowledge

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

INTRODUCTION TO DATA SCIENCE

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

Course Code:25SM2402	**Course Type:** Mandatory Core Course
Programs Offered: All CSE Cluster Programs	**Total Hours:** 30 Hours
Semester: IV	**Hours/Week:** 02 Hours
Credits: 02	**L – T – P – J:** 2 – 0 –0 – 0
Prerequisites: Python, Probability and Statistics	

Course Vision

To provide foundational knowledge of data science concepts, statistical analysis, data pre-processing, and visualization techniques for solving real-world problems across multiple engineering domains.

Course Objectives

Students will be able to:

- Understand fundamental data science concepts and workflows
- Apply statistical techniques and visualization for data analysis
- Perform data cleaning and pre-processing
- Use feature selection and dimensionality reduction methods
- Gain exposure to scalable data analytics techniques

Course Outcomes (CO)

CO	Course Outcomes	Bloom taxonomy level
CO1	Explain basic data science concepts and data preprocessing techniques	L3
CO2	Apply statistical methods and visualization techniques for data analysis	L4
CO3	Perform data cleaning and feature selection for efficient data preparation	L4
CO4	Apply data transformation and dimensionality reduction techniques	L3
CO5	Understand scalable data analytics concepts using modern tools	L5

Broad Course Description

Introduction to Data Science is a 2-credit core course offered in the 4th Semester across all CSE Cluster Programs. It introduces students to the foundational concepts and workflows of data science, including data collection, preprocessing, statistical analysis, and visualization. The course equips students with practical skills in Python-based data science libraries and prepares them for applying data-driven techniques to real-world engineering problems.

Students explore topics such as descriptive statistics, data visualization, data cleaning and feature selection, dimensionality reduction, and accelerated data analytics using GPU-based tools (NVIDIA RAPIDS). Through hands-on programming exercises and a case study on accelerated data processing, the course develops both conceptual understanding and practical competence in handling, analyzing, and interpreting datasets for informed decision-making.

Course Modules

Module 1: Basic Concepts and Python Packages **(6 Hours)**

Topics: Predictive Modelling, 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

Engineering Applications:

Development of predictive analytics systems, industrial data processing applications, healthcare data analysis, automation using Python, smart manufacturing systems, IoT data handling, financial forecasting systems, and scientific computing applications using Python libraries such as NumPy, Pandas, and Scikit-learn.

Module 2: Descriptive Statistics and Data Visualization **(6 Hours)**

Topics: 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

Engineering Applications:

Data-driven decision making in industries, quality control analysis, healthcare statistics and visualization, business intelligence dashboards, manufacturing process optimization, sensor data analysis, correlation-based fault analysis, and engineering performance monitoring using visualization techniques and statistical analysis.

Module 3: Data Cleaning and Feature Selection **(6 Hours)**

Topics: 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

Engineering Applications:

Medical data preprocessing, fraud detection systems, predictive maintenance in industries, preprocessing of IoT sensor data, removal of noisy data in communication systems, feature optimization for machine learning models, image and signal preprocessing, and smart system reliability enhancement.

Module 4: Data Transformation and Dimensionality Reduction **(6 Hours)**

Topics: Scaling Techniques, Min-Max Scaling, Standard Scaling, Handling Outliers, Encoding Categorical Data, Transforming Data Distributions, Feature Engineering, PCA, LDA, SVD Techniques

Engineering Applications:

High-dimensional biomedical data analysis, face recognition systems, image compression, industrial process optimization, intelligent recommendation systems, pattern recognition, speech and signal processing, feature engineering for AI systems, and efficient handling of large-scale engineering datasets.

Module 5: Accelerated Data Analytics (NVIDIA DLI) **(6 Hours)**

Topics: 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

Engineering Applications:

GPU-accelerated machine learning applications, real-time big data analytics, autonomous vehicle data processing, smart city analytics, deep learning acceleration, industrial automation, high-performance computing applications, accelerated scientific simulations, healthcare imaging analytics, and large-scale AI model training using RAPIDS and GPU platforms.

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.
CASE STUDY 1: NVIDIA Course Certification: U Accelerated Data Science - Fundamentals of Accelerated Data Science (Self-Paced Learning)
Link: https://learn.nvidia.com/courses/course detail?course_id=course-v1:DLI+C-DS 02+V2

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)
AI Integration:
Module 1: Students use Python-based data science tools (NumPy, Pandas, Scikit-learn) to load, preprocess, and prepare real-world datasets for AI/ML pipelines. AI tools assist in identifying data quality issues and suggesting preprocessing strategies.
Module 2: AI tools support exploratory data analysis (EDA), automated visualization generation, and statistical pattern recognition in datasets.
Module 3: Students apply AI-assisted imputation tools and automated feature selection methods to clean and prepare datasets for downstream analysis.
Module 4: AI-based dimensionality reduction tools (PCA, LDA) are used to optimize feature spaces for downstream ML applications and improve model efficiency.
Module 5: Students explore GPU-accelerated AI workflows using NVIDIA RAPIDS and DGX systems, running large-scale data analytics pipelines for real-world applications.

Assessment Scheme based on the course

Criteria	Weightage of Marks
Mini Project	10
Continuous Assessment Tool (CAT)	20
Mid-Semester Examination	30
End-Semester Examination	40
Total	**100 marks**

Recommended Tools
The following tools and platforms are recommended to support teaching, learning, and assessment in this course:

Primary Textbooks
Jason Brownlee, "Data Preparation for Machine Learning", Machine Learning Mastery, 2020
1. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022
2. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
3. DLI Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-DS-03+V1

Additional AI & Computational Resources
1. NPTEL: Introduction to Data Science – nptel.ac.in
2. Kaggle Learn (free data science micro-courses) – kaggle.com/learn
3. Google Colab Python Notebooks – colab.research.google.com
4. Scikit-learn Examples and Tutorials – scikit-learn.org
5. NVIDIA DLI Course: Accelerating Data Science Workflows with RAPIDS – learn.nvidia.com/courses/course-detail?course_id=course-v1: DLI+T-DS-03+V1

Project Component (Sample Projects)

Project 1 – **End-to-End Data Analysis on a Real-World Dataset**
Problem: Perform complete data analysis on a provided dataset (sales, health, or sensor data)
Concepts Used: Data preprocessing, EDA, visualization, feature selection
Industrial Relevance: Business analytics, healthcare informatics, IoT data processing

Project 2 – **Customer Churn Prediction using Feature Engineering**
Problem: Identify key features predicting customer churn from telecom data
Concepts Used: Feature selection, encoding, dimensionality reduction, PCA
Industrial Relevance: CRM systems, marketing analytics, customer retention strategies

Project 3 – **Outlier Detection in Industrial Sensor Data**
Problem: Detect anomalies in manufacturing sensor readings for quality control
Concepts Used: IQR method, z-score, KNN imputation, statistical analysis
Industrial Relevance: Predictive maintenance, smart manufacturing, fault detection

Project 4 – **GPU-Accelerated Data Processing using RAPIDS**
Problem: Demonstrate performance comparison of CPU (Pandas) vs GPU (cuDF) operations
Concepts Used: RAPIDS cuDF, dataframe operations, benchmarking
Industrial Relevance: High-performance computing, real-time analytics, big data

Project 5 – **Epidemic Data Analysis and Visualization Dashboard**
Problem: Analyze and visualize population-scale epidemic spread data
Concepts Used: Data merging, normalization, visualization, large-scale analytics

AI-Powered Assignments (Sample Assignments)

Assignment 1: AI-Assisted Data Preprocessing and Quality Evaluation
Students prompt an AI tool: Identify and fix data quality issues (missing values, duplicates, outliers) in this dataset and suggest appropriate preprocessing steps.

- Students must:
    1. Analyse whether the AI-recommended preprocessing steps are statistically justified
    2. Identify any incorrect imputations or inappropriate transformations suggested by the AI
    3. Improve the preprocessing pipeline using standard techniques (IQR, z-score, domain knowledge)
- **Assessment focuses on data quality reasoning, preprocessing decisions, and critical evaluation of AI outputs**

Assignment 2: AI-Assisted Statistical Analysis and Visualization

1. Analyse whether the statistical measures and chart types chosen are appropriate for the data
2. Identify incorrect chart choices, misleading visualizations, or statistical errors

Verify and correct using proper statistical formulas and standard visualization best practices.
Assessment focuses on statistical reasoning, visualization design, and interpretation skills

Assignment 3: AI-Based Feature Selection and Dimensionality Reduction
Students prompt an AI tool: Select the most important features from this dataset and reduce its dimensionality using PCA.

- **Students must:**
  1. Analyse whether the selected features are statistically significant and PCA is applied correctly
  2. Identify incorrect variance thresholds, dropped important features, or over-reduction
  3. Refine using correlation analysis, Recursive Feature Elimination (RFE), and correct PCA implementation

Assessment focuses on feature engineering, PCA interpretation, and model validation.

Assignment 4: AI-Based Accelerated Data Processing Analysis
Students prompt an AI tool: Write a Python script to compare Pandas (CPU) and RAPIDS cuDF (GPU) performance on a large dataset.

- **Students must:**
  1. Analyse whether the benchmarking methodology and performance comparison are valid
  2. Identify incorrect timing methods, unfair comparisons, or hardware constraints overlooked
  3. Improve the benchmarking script using Python time module and correct cuDF API usage

Assessment focuses on GPU computing concepts, performance benchmarking, and critical evaluation of AI-generated code.

CO–PO–PSO Mapping Table

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	3
CO1	3	2	2	–	3	–	–	–	–	–	–	3	2	1
CO2	3	3	2	2	3	–	–	–	1	1	–	3	2	2
CO3	3	3	3	2	3	–	–	–	1	–	–	3	3	2
CO4	3	3	3	3	3	1	–	–	1	1	–	3	3	3
CO5	3	3	3	3	3	1	1	1	1	1	1	3	3	3

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

45 Hours Weekly Lecture Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Predictive Modelling	Lecture + examples	AI-generated predictive modelling examples	Understand predictive modelling concepts
	2	Data Preparation and Importance	Lecture + discussion	AI-assisted preprocessing workflows	Understand importance of data preparation
	3	Data Cleaning Techniques	Lecture + coding demo	AI-assisted noisy data detection	Apply data cleaning methods

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
2	4	Feature Selection Concepts	Lecture + examples	AI-generated feature importance analysis	Understand feature selection techniques
	5	Data Transformation Techniques	Lecture + coding demo	AI-assisted transformation visualization	Apply data transformation methods
	6	Introduction to Python Packages	Lecture + hands-on	AI-assisted Python environment setup	Use NumPy, Pandas, Matplotlib, and Scikit-learn
3	7	Working with NumPy Arrays	Lecture + coding demo	AI-generated array operations	Perform numerical computations using NumPy
	8	DataFrames using Pandas	Lecture + practical session	AI-assisted DataFrame manipulation	Handle datasets using Pandas
	9	Loading and Exploring ML Datasets	Lecture + case study	AI-assisted dataset profiling	Load and analyze machine learning datasets
4	10	Introduction to Descriptive Statistics	Lecture + examples	AI-assisted statistical calculations	Understand statistical measures
	11	Mean, Median, Standard Deviation	Lecture + problem solving	AI-generated statistics examples	Compute statistical parameters
	12	Skewness and Kurtosis	Lecture + visualization	AI-assisted distribution analysis	Analyze data distributions
5	13	Box Plot and Histogram Visualization	Lecture + coding demo	AI-assisted graph generation	Visualize numerical distributions
	14	Scatter Plot and Heat Maps	Lecture + practical session	AI-generated correlation maps	Analyze relationships between variables
	15	Correlation Analysis and Pivot Tables	Lecture + examples	AI-assisted business intelligence dashboards	Perform correlation analysis
6	16	ANOVA and Data Analysis Process	Lecture + discussion	AI-assisted variance analysis	Understand ANOVA applications
	17	Data Rescaling and Standardization	Lecture + coding demo	AI-assisted scaling comparison	Apply scaling and normalization
	18	Normalization and Binarization	Lecture + problem solving	AI-generated preprocessing workflows	Perform normalization and binarization
7	19	Univariate and Bivariate Analysis	Lecture + examples	AI-assisted pattern analysis	Analyze single and paired variables
	20	Recursive Feature	Lecture +	AI-generated feature	Apply feature

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Elimination	coding demo	ranking	elimination methods
	21	Principal Component Analysis (PCA)	Lecture + visualization	AI-assisted dimensionality reduction	Understand PCA concepts
8	22	Handling Missing Data	Lecture + examples	AI-assisted missing value detection	Handle incomplete datasets
	23	Outlier Detection and Removal	Lecture + coding demo	AI-generated anomaly detection	Detect and remove outliers
	24	Statistical Imputation Methods	Lecture + practical session	AI-assisted imputation analysis	Apply statistical imputation
9	25	KNN Imputation	Lecture + coding demo	AI-assisted KNN visualization	Implement KNN-based imputation
	26	Iterative Imputation Techniques	Lecture + examples	AI-assisted iterative optimization	Apply iterative imputation methods
	27	Feature Selection for Numerical and Categorical Data	Lecture + discussion	AI-assisted feature optimization	Select features for ML models
10	28	Scaling Techniques and Min-Max Scaling	Lecture + coding demo	AI-generated scaling comparison	Apply Min-Max scaling
	29	Standard Scaling and Outlier Handling	Lecture + practical session	AI-assisted preprocessing pipelines	Apply standard scaling methods
	30	Encoding Categorical Data	Lecture + examples	AI-generated encoding workflows	Encode categorical variables
11	31	Transforming Data Distributions	Lecture + discussion	AI-assisted transformation analysis	Transform skewed datasets
	32	Feature Engineering Concepts	Lecture + coding demo	AI-assisted feature generation	Perform feature engineering
	33	Linear Discriminant Analysis (LDA)	Lecture + visualization	AI-assisted classification analysis	Understand LDA for classification
12	34	Singular Value Decomposition (SVD)	Lecture + examples	AI-assisted matrix decomposition	Apply SVD techniques
	35	High-Dimensional Data Analysis	Lecture + case study	AI-assisted biomedical data analysis	Handle large-scale datasets
	36	Image Compression and Pattern Recognition	Lecture + coding demo	AI-generated image compression	Apply dimensionality reduction in imaging
13	37	CPU vs GPU in Data Processing	Lecture + discussion	AI-assisted hardware comparison	Compare CPU and GPU computing

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	38	Introduction to RAPIDS	Lecture + practical session	AI-assisted RAPIDS workflows	Understand GPU-accelerated analytics
	39	GPU-Accelerated DataFrames	Lecture + coding demo	AI-generated GPU analytics examples	Perform accelerated DataFrame operations
14	40	Machine Learning Acceleration using GPUs	Lecture + examples	AI-assisted ML acceleration	Apply GPU acceleration in ML
	41	Identifying Processing Bottlenecks	Lecture + case study	AI-assisted bottleneck analysis	Analyze system performance bottlenecks
	42	Real-Time Big Data Analytics	Lecture + discussion	AI-assisted streaming analytics	Understand real-time analytics systems
15	43	Healthcare Imaging Analytics using GPUs	Lecture + case study	AI-assisted healthcare image analysis	Apply accelerated analytics in healthcare
	44	NVIDIA DGX Case Study for Epidemic Control	Lecture + discussion	AI-assisted large-scale analytics	Analyze population-scale data systems
	45	Course Review and Problem Solving	Q&A + revision	AI-generated assessment	Consolidate all data analytics concepts

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

IOT & SENSOR SYSTEMS

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

Course Code:25SM2403	**Course Type:** Integrated Program Core Course
Programs Offered: CSME	**Total Hours:** 45 Hours
Semester: IV	**Hours/Week:** 03 Hours
Credits: 03	**L – T – P – J:** 3 – 0 –0 – 0
Prerequisites: Internet of Things, Data Analytics, and Basic Knowledge of Computer Architecture	

Course Vision

To develop the ability to design intelligent IoT-based sensor systems by integrating real-time data acquisition, sensor analytics, and GPU-accelerated computing, enabling efficient, scalable, and high-performance smart systems.

Course Objectives:

Students will be able to:

1. Understand IoT architectures, embedded platforms such as Arduino and Raspberry Pi, and sensor interfacing techniques
2. Analyze sensor data using statistical and machine learning methods
3. Apply sensor analytics for real-time data processing and decision making
4. Evaluate system performance, latency, and scalability in IoT systems
5. Design intelligent systems integrating IoT, sensor analytics, and GPU-based acceleration

Course Outcomes (CO)

CO	Outcome	Bloom taxonomy level
CO1	Explain IoT systems, embedded platforms, and sensor technologies	L2
CO2	Analyze sensor data using analytics and machine learning techniques	L4
CO3	Apply sensor analytics for real-time processing and decision making	L3
CO4	Evaluate system performance, scalability, and data efficiency	L5
CO5	Design intelligent IoT systems integrated with GPU-based acceleration	L5

Broad Course Description

This course provides a comprehensive understanding of IoT systems and sensor analytics, focusing on data acquisition, processing, and intelligent decision-making. It introduces embedded platforms such as Arduino and Raspberry Pi for building real-world IoT systems and explores how sensor data is collected, processed, and analyzed.

The course further integrates advanced sensor analytics techniques including machine learning, time-series analysis, and real-time processing. It also introduces GPU-based acceleration using NVIDIA platforms to enable high-performance data processing and AI-driven IoT applications.

Course Modules

Module - 1: IoT Systems and Embedded Platforms (9 Hours)

Introduction to Internet of Things, IoT architecture and components, embedded systems for IoT, Arduino architecture and programming, Raspberry Pi architecture and interfacing, sensor interfacing with Arduino and Raspberry Pi, IoT communication protocols MQTT HTTP CoAP, wireless communication technologies, real-time data acquisition, IoT system design principles.

Engineering Applications:

Smart home automation systems, industrial IoT monitoring, wearable healthcare devices, smart agriculture systems, environmental monitoring stations, embedded control systems, smart energy management, connected transportation systems, robotics and automation applications.

Module - 2: IoT Data Acquisition and Communication (9 Hours)

Sensor types and classification, analog and digital sensors, sensor calibration and characteristics, signal conditioning, data acquisition systems, sampling and quantization, noise filtering techniques, wireless sensor networks, IoT gateways, cloud connectivity, data transmission protocols, real-time IoT communication systems.

Engineering Applications:

Real-time industrial process monitoring, wireless health monitoring systems, smart city infrastructure, remote sensing applications, intelligent traffic management, structural health monitoring, smart water management, factory automation and predictive maintenance systems.

Module - 3: Sensor Data Analytics Fundamentals (9 Hours)

Sensor data characteristics, data preprocessing and cleaning, normalization techniques, feature extraction methods, statistical analysis of sensor data, time series analysis, anomaly detection techniques, data visualization, data fusion from multiple sensors.

Engineering Applications:

Fault detection in industrial equipment, healthcare sensor data analysis, environmental pollution monitoring, smart grid analytics, autonomous vehicle sensing systems, biometric signal analysis, energy consumption analysis, multi-sensor fusion in robotics and surveillance systems.

Module - 4: Advanced Sensor Analytics and Machine Learning (9 Hours)

Machine learning for sensor data, classification and clustering techniques, predictive analytics, real-time data analytics frameworks, model deployment strategies, lightweight machine learning models, optimization techniques, latency reduction, scalable analytics systems.

Engineering Applications:

Predictive maintenance in manufacturing, AI-based healthcare diagnostics, intelligent transportation systems, smart surveillance and security, anomaly detection in cyber-physical systems, industrial automation using AI, autonomous drones and robotics, edge AI applications for real-time decision making.

Module - 5: GPU Architecture and Real-Time Analytics (9 Hours)

Introduction to GPU computing, CUDA fundamentals, parallel processing concepts, GPU memory hierarchy, NVIDIA Jetson platform, sensor data processing using GPU, real-time analytics using GPU acceleration, performance optimization techniques, deploying AI models on edge devices.

Engineering Applications:

High-speed video analytics, autonomous vehicle processing systems, medical image processing, GPU-accelerated AI applications, real-time object detection systems, smart edge computing devices, robotics vision systems, deep learning deployment on embedded platforms such as NVIDIA Jetson for industrial and healthcare applications.

CASE STUDIES: DLI Learning Path
Integrating Sensors with NVIDIA DRIVE (Self-Paced Learning)
NVIDIA Course Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-AV-06+V1
Build High-Performance and AI-Enabled Sensor Processing Applications (Self-Paced Learning)
NVIDIA Course Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-30+V1

AI Integration:
Module 1: Students use Arduino and Raspberry Pi platforms to collect and process real-time sensor data.
Module 2: Simulation tools are used for analyzing sensor signals and communication protocols.
Module 3: AI tools are used for data preprocessing, visualization, and anomaly detection.
Module 4: Machine learning models are applied for predictive analytics on sensor data.
Module 5: GPU-based tools and CUDA environments are used for high-performance data processing and AI acceleration.

Assessment Scheme based on the course

Criteria	Weightage of Marks
Class Tests / Assignment	10
Mid-Semester Examination	30
Hackathon	10
Project	10
SEE	40
Total	100 marks

Recommended Tools
1. Python (for data processing, analytics, and machine learning applications) TensorFlow or PyTorch (for building and deploying machine learning models for sensor analytics)
2. Arduino (for IoT prototyping, sensor interfacing, and embedded system development)
3. Raspberry Pi (for edge computing, IoT system implementation, and real-time data processing)
4. MQTT and IoT Communication Platforms (for lightweight data transmission and device communication)
5. NVIDIA Jetson Platform (for edge AI deployment and real-time GPU-based processing)
6. CUDA Programming Tools (for parallel computing and GPU acceleration of sensor data processing)

Primary Textbooks (Full References)
1. J. S. Wilson, Sensor Technology Handbook, 1st ed., Burlington, MA, USA: Elsevier, 2005.
2. P. Lea, Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security, Birmingham, UK: Packt Publishing, 2018.
3. A. Tamboli, Build Your Own IoT Platform: Develop a Fully Flexible and Scalable Internet of Things Platform in 24 Hours, Birmingham, UK: Packt Publishing, 2019.
4. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.

Mini Project Component

The following projects are suggested based on course outcomes. Students may attempt one project individually or in pairs.

Project 1 – Smart Health Monitoring System

Problem: Develop an IoT-based system to monitor patient health parameters in real time Concepts Used: Sensors, Arduino/Raspberry Pi, data acquisition, real-time analytics Industrial Relevance: Healthcare systems, wearable devices, remote patient monitoring

Project 2 – Real-Time Traffic Monitoring System

Problem: Design a system to monitor traffic conditions using IoT sensors and analytics Concepts Used: Sensor networks, data transmission, real-time analytics, visualization Industrial Relevance: Smart cities, transportation systems, traffic management

Project 3 – Industrial Predictive Maintenance System

Problem: Predict machine failures using sensor data and analytics Concepts Used: Sensor data analysis, machine learning, anomaly detection Industrial Relevance: Manufacturing industries, automation, Industry 4.0

Project 4 – Sensor-Based Anomaly Detection System

Problem: Detect abnormal patterns in sensor data for safety and monitoring Concepts Used: Data preprocessing, machine learning, real-time analytics Industrial Relevance: Security systems, industrial monitoring, environmental safety

Project 5 – GPU-Accelerated IoT Analytics System

Problem: Implement high-speed sensor data processing using GPU acceleration Concepts Used: CUDA programming, parallel processing, performance optimization Industrial Relevance: AI systems, high-performance computing, real-time analytics

AI-Powered Assignments

Assignment 1: AI-Based IoT System Design using Arduino and Raspberry Pi

Students prompt an AI tool: Design an IoT system using Arduino and Raspberry Pi for real-time data acquisition. Students must:

- Analyse the correctness of system architecture and hardware interfacing
- Identify errors in sensor integration, communication protocols, or data flow
- Validate the design using simulation tools or prototype implementation
- Assessment focuses on IoT architecture understanding, embedded system design, and validation of AI-generated outputs

Assignment 2: AI-Based Sensor Data Preprocessing and Analysis

Students prompt an AI tool: Clean and preprocess raw sensor data and perform statistical analysis. Students must:

- Analyse correctness of preprocessing techniques such as normalization and filtering
- Identify issues like noise, missing values, or incorrect transformations
- Validate results using standard data analysis tools
- Assessment focuses on data preprocessing skills, analytical accuracy, and validation ability

Assignment 3: AI-Based Time-Series Sensor Data Prediction

Students prompt an AI tool: Predict future values of sensor data using time-series models. Students must:

- Analyse correctness of prediction models and output trends
- Identify overfitting, underfitting, or incorrect assumptions
- Validate results using visualization and performance metrics
- Assessment focuses on predictive analytics, time-series understanding, and evaluation of AI outputs

Assignment 4: AI-Based Anomaly Detection in Sensor Data

Students prompt an AI tool: Detect anomalies in sensor data using machine learning techniques. Students must:

- Analyse correctness of anomaly detection results
- Identify false positives and missed anomalies
- Validate using datasets and performance metrics
- Assessment focuses on machine learning application, anomaly detection, and critical evaluation

Assignment 5: GPU-Based Sensor Data Processing using CUDA

Students prompt an AI tool: Implement parallel processing of sensor data using GPU acceleration. Students must:

- Analyse performance improvements over CPU-based processing
- Identify inefficiencies in parallel execution
- Validate results using CUDA-based tools or benchmarking

CO-PO mapping

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	3
CO1	3	2	-	-	3	1	-	-	-	-	-	2	2	2
CO2	3	3	-	-	3	1	-	-	-	-	1	2	2	2
CO3	3	3	2	-	3	1	-	-	-	-	2	2	2	2
CO4	3	3	2	-	3	1	-	1	1	-	2	3	2	2
CO5	3	3	2	-	3	1	-	2	2	-	2	3	2	2

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

Week-by-Week Teaching Plan (45 Lectures)

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Sets and Subsets	Lecture + examples	AI-generated set examples	Understand basic concepts of sets and subsets
	2	Operations on Sets	Lecture + discussion	AI-assisted set operation visualization	Perform union, intersection, and complement operations
	3	Algebraic Properties of Sets and Addition Principle	Lecture + problem solving	AI-generated counting examples	Apply algebraic properties and counting principles
2	4	Introduction to Relations and Properties	Lecture + examples	AI-generated relation examples	Define relations and their properties
	5	N-ary Relations and Applications	Lecture + discussion	AI-assisted relational mapping	Analyze N-ary relations in engineering applications
	6	Representing Relations	Lecture + coding demo	AI-generated matrix representation	Represent relations using matrices and graphs

3	7	Closure of Relations	Lecture + examples	AI-assisted closure computation	Compute reflexive, symmetric, and transitive closures
	8	Equivalence Relations	Lecture + problem solving	AI-generated equivalence examples	Identify and analyze equivalence relations
	9	Partial Order Relations	Lecture + discussion	AI-assisted ordering visualization	Understand partially ordered sets
4	10	Functions and Their Types	Lecture + examples	AI-assisted function classification	Define functions and identify their types
	11	Growth of Functions	Lecture + problem solving	AI-generated complexity analysis	Analyze asymptotic growth of functions
	12	Warshall's Algorithm and Transitive Closure	Lecture + coding demo	AI-assisted path computation	Apply Warshall's algorithm for transitive closure
5	13	Introduction to Mathematical Induction	Lecture + examples	AI-generated induction examples	Understand principles of mathematical induction
	14	Recursive Relations and Fibonacci Numbers	Lecture + discussion	AI-assisted recurrence visualization	Solve Fibonacci and recurrence problems
	15	Tower of Hanoi Problem	Lecture + problem solving	AI-generated recursion steps	Apply recursion concepts to classical problems
6	16	Code Word Enumeration	Lecture + examples	AI-assisted combinatorial generation	Analyze coding and enumeration problems
	17	Solving Linear Recurrence Relations	Lecture + problem solving	AI-generated recurrence solutions	Solve linear recurrence equations
	18	Truth Tables and Logical Connectives	Lecture + coding demo	AI-assisted logical evaluation	Construct truth tables and evaluate logical expressions
7	19	Introduction to Graph Theory	Lecture + examples	AI-generated graph examples	Understand graph terminology and models
	20	Types of Graphs and Graph Terminology	Lecture + discussion	AI-assisted graph classification	Identify special graphs and their properties
	21	Bipartite and Complete Bipartite Graphs	Lecture + problem solving	AI-generated bipartite visualizations	Analyze bipartite graph structures
8	22	Graph Representations	Lecture + coding demo	AI-assisted graph representation	Represent graphs using adjacency matrices and lists
	23	Graph Isomorphism	Lecture + examples	AI-generated graph comparison	Determine graph isomorphism

	24	Paths and Connectivity in Graphs	Lecture + problem solving	AI-assisted connectivity analysis	Analyze connected graphs and paths
9	25	Directed and Undirected Graph Connectivity	Lecture + examples	AI-generated connectivity examples	Understand connectivity in directed and undirected graphs
	26	Vertex and Edge Connectivity	Lecture + discussion	AI-assisted network reliability analysis	Analyze vertex and edge connectivity
	27	Applications of Graph Connectivity	Lecture + case studies	AI-generated network applications	Apply connectivity concepts to engineering problems
10	28	Euler Paths and Circuits	Lecture + examples	AI-assisted Euler path visualization	Identify Eulerian paths and circuits
	29	Hamiltonian Paths and Circuits	Lecture + discussion	AI-generated Hamiltonian examples	Analyze Hamiltonian paths and circuits
	30	Applications of Euler and Hamilton Graphs	Lecture + problem solving	AI-assisted routing applications	Apply graph traversal concepts in real-world systems
11	31	Planar Graphs	Lecture + examples	AI-generated planar graph visualization	Understand planar graph properties
	32	Applications of Planar Graphs	Lecture + case study	AI-assisted layout optimization	Analyze planar graph applications
	33	Graph Coloring Concepts	Lecture + coding demo	AI-assisted coloring visualization	Understand graph coloring techniques
12	34	Chromatic Number and Graph Coloring Applications	Lecture + problem solving	AI-generated scheduling applications	Apply graph coloring to scheduling problems
	35	Medical Imaging Applications of Graph Theory	Lecture + examples	AI-assisted medical image analysis	Apply graph theory in healthcare imaging
	36	Object Detection and Facial Recognition Applications	Lecture + discussion	AI-generated computer vision examples	Understand graph applications in AI vision systems
13	37	Satellite Image Analysis and Disaster Monitoring	Lecture + case study	AI-assisted satellite analytics	Analyze graph-based remote sensing applications
	38	Diabetic Retinopathy Detection	Lecture + examples	AI-assisted retinal image classification	Understand AI-driven healthcare diagnostics
	39	Brain Tumor Segmentation using MRI	Lecture + coding demo	AI-generated segmentation visualization	Analyze graph-based medical image segmentation

14	40	Autoencoders for Medical Image Denoising	Lecture + examples	AI-assisted denoising models	Apply autoencoders for image enhancement
	41	GANs for Synthetic Medical Image Generation	Lecture + discussion	AI-generated synthetic images	Understand GAN-based medical augmentation
	42	Anomaly Detection in Industrial IoT	Lecture + problem solving	AI-assisted anomaly detection	Apply graph methods in IoT anomaly detection
15	43	Drug Discovery and Molecular Generation	Lecture + case study	AI-assisted molecular generation	Analyze graph applications in drug discovery
	44	Recommendation Systems using Autoencoders	Lecture + coding demo	AI-generated recommendation models	Understand recommendation systems using AI
	45	Course Review and Problem Solving	Q&A session	AI-generated practice problems	Consolidate concepts of discrete mathematics and graph applications

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

MACHINE LEARNING

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2404	**Course Type:** Mandatory Core Course
Programs Offered: CSE, CSME, AI&DS, CST	**Total Hours:** 45 + 30 Hours
Semester: IV	**Hours/Week:** 05 Hours
Credits: 04	**L – T – P – J:** 3 – 0 –2 – 0
Prerequisites: Linear Algebra & Data Foundations, Calculus & Mathematical Modelling, Discrete Mathematics & Graph Theory	

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.

Course Outcomes (CO)

CO	Course Outcomes	Bloom taxonomy level
CO1	Describe the basic concepts and different types of Machine Learning techniques and to design Well Posed Learning System.	L3
CO2	Apply the learned concepts of the Supervised learning algorithms for regression and classification problems.	L4
CO3	Apply Unsupervised Machine Learning Algorithms such as Hard and Soft clustering and Dimensionality Reduction techniques for solving appropriate real-world applications.	L4
CO4	Make Use of the learned concepts of the Probabilistic Graphical Models and Ensemble Learning Methods for handling complex real time problems.	L3
CO5	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

Course Modules

Module 1: Foundations of Machine Learning **[09 Hours]**

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 2: Supervised Machine Learning Algorithms [09 Hours]
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 3: Unsupervised Machine Learning Algorithms [09 Hours]
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 4: Probabilistic Graphical Models and Ensemble Learning [09 Hours]
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 5: Machine Learning Model Validation, Optimization, and Regularization [09 Hours]
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.

CASE STUDY 1: NVIDIA Course Certification: Deep Learning Exploring Adversarial Machine Learning (Self-Paced Learning)
Associate- Gen AI LLMs
Link: https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS 03+V1

Department of Computer Science & Medical Engineering

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 suitable Dataset (e.g: 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 House Prices using the Housing Price Dataset.	Google Colab/ Jupyter	2 h	CO2
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 perform classification on suitable dataet (e.g 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 perform Classification (e.g. Wine Quality using the Wine Quality Dataset)	Google Colab/ Jupyter	2 h	CO2
8	Random Forest Model	Develop a Random Forest Model to perform Prediction. (e.g: 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. (e.g 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. (e.g: using the Wine Dataset)	Google Colab/ Jupyter	2 h	CO3

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
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 dataset (e.g Breast Cancer Dataset)	Google Colab/ Jupyter	2 h	CO5
14	Hyperparameter Tuning	Perform Hyperparameter Tuning using Grid Search with Cross-Validation on suitable dataset (e.g: The Digits Dataset)	Google Colab/ Jupyter	2 h	CO5
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

Teaching Learning Process

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

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **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.

- **Authentic Intelligence Reflection**
  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.

RECOMMENDED TOOLS

1. **Programming:** Python (NumPy, Pandas, Scikit-learn, PyTorch) – Google Colab/ Jupyter Notebook/Visual Studio
2. **Visualization:** Matplotlib, Seaborn, Plotly, UMAP
3. **Experiment Tracking:** MLflow, Weights & Biases
4. **MLOps& Deployment:** Docker, FastAPI, ONNX
5. **AI Assistants:** GitHub Copilot, ChatGPT, Claude

SUGGESTED READING

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

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**	**3**
CO1	3	2	1	–	2	–	–	–	–	–	–	3	2	1
CO2	3	3	3	2	3	–	–	–	1	1	–	3	3	2
CO3	3	3	3	2	3	–	–	–	1	–	–	3	3	2
CO4	3	3	3	3	3	1	–	–	1	1	1	3	3	3
CO5	3	3	3	3	3	1	1	1	1	1	1	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Week-by-Week Teaching Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Machine Learning and Learning Systems	Lecture + Discussion	AI-generated ML application examples	Understand fundamentals of machine learning
	2	Well-posed Learning Problems	Lecture + Examples	AI-assisted concept visualization	Identify characteristics of learning problems
	3	Types of Learning: Supervised, Unsupervised, Reinforcement	Lecture + PPT	AI-generated comparison charts	Differentiate ML paradigms
2	4	Data Preprocessing Techniques	Lecture + Demo	AI-assisted preprocessing tools	Understand preprocessing importance
	5	Missing Value Imputation	Problem Solving	AI-generated imputation examples	Handle incomplete datasets
	6	Balanced and Imbalanced Data Handling	Lecture + Case Study	AI-assisted resampling visualization	Apply balancing techniques
3	7	Feature Engineering Concepts	Lecture + Examples	AI-generated feature extraction ideas	Perform feature engineering
3	8	Real-world ML Applications	Discussion + Case Study	AI-generated predictive analytics examples	Relate ML to industry applications
	9	Introduction to Supervised Learning	Lecture + Demo	AI-assisted classification visualization	Understand supervised learning workflow
4	10	Binary and Multi-Class Classification	Lecture + Practice	AI-generated classification datasets	Solve classification problems

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	11	Linear Regression	Lecture + Coding Demo	AI-assisted regression plotting	Apply linear regression
	12	Nonlinear Regression	Lecture + Numerical Examples	AI-generated nonlinear fitting	Model nonlinear relationships
5	13	Logistic Regression	Lecture + Coding	AI-assisted probability visualization	Implement logistic regression
	14	Naive Bayes Classification	Lecture + Examples	AI-generated Bayesian inference	Apply Naive Bayes classifier
	15	Decision Trees (ID3, CART)	Lecture + Diagram	AI-generated decision tree construction	Build decision tree models
6	16	Random Forest Algorithm	Lecture + Demo	AI-assisted ensemble visualization	Understand ensemble learning basics
	17	K-Nearest Neighbor Algorithm	Problem Solving	AI-generated distance computation	Apply KNN classification
	18	Support Vector Machines	Lecture + Visualization	AI-assisted hyperplane visualization	Implement SVM models
7	19	Introduction to Unsupervised Learning	Lecture + Discussion	AI-generated clustering examples	Understand unsupervised learning
	20	Clustering Concepts	Lecture + Examples	AI-assisted cluster visualization	Identify clustering techniques
	21	K-Means Clustering	Coding Demo	AI-generated centroid updates	Implement K-Means algorithm
8	22	Hierarchical Clustering	Lecture + Practice	AI-generated dendrogram visualization	Apply hierarchical clustering
	23	Dimensionality Reduction Concepts	Lecture + Discussion	AI-assisted PCA visualization	Understand dimensionality reduction
	24	Principal Component Analysis (PCA)	Lecture + Numerical Examples	AI-generated feature projection	Perform PCA
9	25	Association Rule Mining	Lecture + Case Study	AI-generated market basket analysis	Apply association mining
	26	Anomaly Detection	Lecture + Examples	AI-assisted anomaly visualization	Detect outliers in datasets

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
	27	Applications of Unsupervised Learning	Discussion	AI-generated real-world use cases	Relate unsupervised learning to applications
10	28	Bayesian Networks	Lecture + Visualization	AI-assisted probabilistic graphs	Understand Bayesian reasoning
	29	Markov Random Fields	Lecture + Examples	AI-generated graphical model diagrams	Analyze probabilistic dependencies
	30	Ensemble Learning Concepts	Lecture + Discussion	AI-assisted ensemble comparisons	Understand ensemble techniques
11	31	Bagging Algorithms	Lecture + Coding Demo	AI-generated bootstrap sampling	Implement bagging methods
	32	Boosting Techniques: AdaBoost	Lecture + Examples	AI-assisted boosting flowcharts	Apply AdaBoost
	33	Gradient Boosting Machines	Lecture + Demo	AI-generated boosting visualization	Understand GBM workflow
12	34	XGBoost and CatBoost	Lecture + Case Study	AI-assisted model comparison	Analyze advanced boosting algorithms
	35	Out-of-Bag Score	Lecture + Examples	AI-generated validation metrics	Evaluate ensemble models
	36	Applications of Ensemble Methods	Discussion	AI-assisted fraud detection examples	Apply ensemble methods in real-world tasks
13	37	Hyperparameter Tuning	Lecture + Coding	AI-assisted parameter optimization	Tune ML model parameters
	38	Grid Search CV and Random Search CV	Coding Demo	AI-generated search space visualization	Apply cross-validation techniques
	39	Confusion Matrix and Accuracy Metrics	Lecture + Practice	AI-assisted metric calculations	Evaluate classification performance
14	40	Precision, Recall, F1 Score	Problem Solving	AI-generated performance comparisons	Analyze evaluation metrics
	41	Error Metrics: MSE, RMSE, MAE	Lecture + Numerical Examples	AI-assisted regression analysis	Evaluate regression models
	42	Overfitting, Underfitting, Bias and Variance	Lecture + Visualization	AI-generated fitting curves	Analyze model generalization

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
15	43	Regularization Techniques (L1, L2, Elastic Net)	Lecture + Coding Demo	AI-assisted regularization comparison	Apply regularization methods
	44	Optimization Algorithms and Bayesian Optimization	Lecture + Discussion	AI-generated optimization flowcharts	Understand ML optimization
	45	Course Revision and Final Problem Solving	Interactive Session	AI-generated quizzes and case studies	Integrate and apply complete ML concepts

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DIGITAL LOGIC DESIGN & COMPUTER ARCHITECTURE

[As per Choice based credit system (CBCS) Scheme]

Course Code:25SM2405	**Course Type:** Program Core Course
Programs Offered: CSME & CSE	**Total Hours:** 45 Hours(L)+ 30 Hours(P)
Semester: IV	**Hours/Week:** 05 Hours
Credits: 04	**L – T – P – J:** 3 – 0 – 2 – 0
Prerequisites: Number Systems, Boolean Logic Basics, C	

Course Vision

To equip students with strong foundational and practical knowledge in digital logic design and hardware modeling using Verilog, enabling them to design, analyze, and implement efficient digital systems aligned with modern VLSI, FPGA, and AI-driven hardware applications.

Course Objectives

Students will be able to:

1. Understand the elements of digital logic functions to digital system abstractions using Verilog.
2. Illustrate the simplification of Boolean expressions using Karnaugh.
3. Use combinational logic circuits for arithmetic operations and logical operations.
4. Interpret and model sequential elements, flip-flops, counter, shift registers.
5. Understand the concepts of processor architecture and instruction set architecture.
6. Understand the implementation of a multi-cycle processor architecture along with optimizations applicable to cache memory.

Course Outcomes (CO)

At the end of course, students will be able to:

CO	Course Outcomes	Bloom taxonomy level
CO1	Interpretation of Boolean Expressions of digital design in simplified form.	L2
CO2	Build the various elements of digital logic system with Verilog.	L3
CO3	Construct Combinational and Sequential logic circuits and analyse the hardware model of a digital system at different levels of abstraction in Verilog	L5
CO4	Understand the basic principles of the quantitative design of computers and Instruction set architecture.	L2
CO5	Analyse the implementation of pipelined processors and cache optimization techniques.	L4

Broad Course Description

This course introduces the fundamentals of digital logic design, including number systems, Boolean algebra, and logic simplification techniques. It covers the design and modeling of combinational and sequential circuits using Verilog HDL. Students learn about flip-flops, counters, finite state machines, and hardware abstraction levels. The course also explores FPGA-based design flows and real-world applications such as ALU and digital systems. Emphasis is placed on simulation, verification, and practical implementation using modern electronic design automation tools.

Detailed Course Syllabus

The course consists of five modules. Each module integrates:

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

Course Modules

Module 1: Introduction **[09 hours]**

INTRODUCTION: Number System- Binary, Hexa, Decimal, Octal and its conversion, Addition and subtraction using 1's and 2's complement, Canonical Notation - SOP & POS forms, Minimization of SOP and POS forms. ARITHMETIC CIRCUITS AND VERILOG MODELLING Adders: Half adder, full adder, Ripple carry adder, parallel adder /subtractor, fast adders-CLA, comparator- 2 bit. Simplification using K-Maps.

NVIDIA DLI Mapping- Introduction to GPU Programming model, CPU vs GPU algorithm, GPU architecture, Basics of CUDA program.

Module 2: Combinational Circuit Building **[09 hours]**

Combinational Circuit Building: Multiplexers 4:1, 8:1, decoders 3:8, 2:4, demultiplexers 1:4, encoders 8:3, 4:2, code converters- B to G and G to B- Simplification using K-Maps.

Verilog for combinational circuits: Behavioural modelling- if else, case-casex, casez, structural modelling.

Module 3: Sequential Circuits-1 **[09 Hours]**

Basic Latch, Gated latches, Flip Flops SR, D, JK, T, Race round condition, master-slave flip-flops JK, 0's and 1's Catching Problem, Conversion of flip-flops SR-JK, JK-T, shift registers- SISO, SIPO, PISO, PIPO, Universal Shift Register, Setup time, Hold time, Propagation delay.

Module 4: Sequential Circuits-2 **[09 hours]**

Binary counters– asynchronous and synchronous, mod-n counter, Verilog blocking and non-blocking, Mealy Model, Moore Model, State Machine Notation, Construction of Finite State Machine.

Module 5: Introduction to Electronic Design Automation and Applications **[09 hours]**

Introduction to Electronic Design Automation: FPGA Design Flow, ASIC Design flow, architectural design, logic design, simulation, verification and testing, 3000 Series FPGA architecture.

Applications: Design 4 Bit ALU, 7 Segment display, Vending Machine, 3 Pipeline.

NVIDIA DLI Course Mapped:

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

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

- **AI Integration:**
  - **Module 2:**Leverage theNVIDIA CUDA Compiler (NVCC) in conjunction with AI-driven optimization likeChatGPT for intelligent code generation, GitHub Copilot for real-time Verilog/CUDA coding assistance, to simulate combinational circuits in parallel, enabling accelerated computation and intelligent workload distribution across GPU cores.
  - **Module 5:** Integrating the NVIDIA CUDA Compiler (NVCC) with AI-powered tools such as ChatGPT revolutionizes EDA workflows by enabling rapid, parallel simulation and intelligent

verification of digital circuits — reducing design cycle time while enhancing accuracy through AI-assisted analysis and real-time feedback.

- **Authentic Intelligence Reflection:**
  - **Module 2:** NVCC amplifies the depth of combinational circuit learning by enabling fast, parallel simulation and validation — while Verilog, guided by AI-assisted design principles such as machine learning-based logic optimization, LLM-driven code generation (e.g., ChatGPT, GitHub Copilot), and neural network-based error detection, ensures precise, reliable, and contextually informed hardware design that bridges theoretical understanding with real-world application.
  - **Module 5:** By applying authentic intelligence encompassing critical human reasoning, AI-powered verification tools (such as ChatGPT for logic validation, DeepMind's AlphaCode for code synthesis, and ML-based formal verification engines), learners develop a deeper comprehension of how GPU-accelerated CUDA simulation, combined with AI-driven Electronic Design Automation (EDA) tools like Synopsys DSO.ai, meaningfully support and strengthen actual hardware design, synthesis, timing analysis, and informed decision-making within professional EDA environments.

Laboratory Component

Part A: Implementation using CUDA NVCC Compiler.

1	Introduction to the CUDA C, a simple CUDA C program to implement basic gates.
2	Design and realize Half and Full Adder using CUDA C.
3	Design and implement the following Combinational designs – I, a. Multiplexer: 4:1, 8:1 MUX. b. De Multiplexer: 1:4, 1:8 DEMUX usingCUDA C program for parallel processing.
4	Design and implement the following Combinational designs – II, c. Encoder with and without Priority: 8:3 and 4:2. d. Decoder: 3:8 and 2:4 usingCUDA C program for parallel processing.

Part B: Implementation using Xilinx

1	Introduction to Verilog, Syntax of Verilog coding, Modelling styles in Verilog, Verilog Operators, Test bench for simulation.
2	Design and realize Half and Full Adder.
3	Design and implement the following Combinational designs – I, a. Multiplexer: 4:1, 8:1 MUX. b. De Multiplexer: 1:4, 1:8 DEMUX
4	Design and implement the following Combinational designs – II, c. Encoder with and without Priority: 8:3 and 4:2. d. Decoder: 3:8 and 2:4
5	Design a 4 bit ALU.
6	Implement D, T and JK Flip Flop.
7	Design of Mod – n Up/Down Counter with Synchronous reset.
8	Design of Mod – n Up/Down Counter with Asynchronous.

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)**: 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.

A. Integrated Professional Core Courses (Theory + Practical)

Criteria	Weightage of Marks
Continuous Assessment Tool (CAT) every 2 weeks (Minimum 06 assessments including AI-powered Assignment)	20
Mid Semester Examination (MSE)	20
Lab Record	05
Lab Examination	15
Total	**60 marks**

Recommended Tools

- **Programming:** Verilog
- **Platforms:** Vivado 2019.1
- **AI Tools:** CUDA NVCC, Circuit Mind

Suggested Reading

Primary Textbooks

1. M. Morris Mano Michael D. Ciletti, "Digital Design with an Introduction to the Verilog HDL", 6th Edition, Pearson Education, 2014.
2. Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog design", McGraw Hill, 2014.
3. Douglas J Smith, "HDL Chip Design", Doone publications 1996.

Additional Resources

1. Nazein M. Botros, "HDL programming (VHDL and Verilog)", Dreamtech Press, 2006.
2. John M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2014.
3. Donald D. Givone, "Digital Principles and Design", McGraw Hill, 2015.
4. Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Pearson Education, 2016.

CO–PO & PSO Mapping Table

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**	**3**
CO1	3	2	-	-	-	-	-	-	-	-	2	-	2	2
CO2	3	2	-	-	-	-	-	-	-	-	2	-	3	3
CO3	3	3	2	2	-	-	-	-	-	-	2	-	3	3
CO4	3	3	3	2	2	-	-	-	-	-	2	-	1	2
CO5	3	3	2	2	-	-	-	-	-	-	2	-	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Week-by-Week Teaching Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to Digital Image Processing	Lecture + examples	AI-assisted concept explanation	Understand the basics of digital image processing
	2	Origins and applications of digital image processing	Lecture + discussion	AI-generated real-world examples	Understand the scope and uses of image processing
	3	Fundamental steps in digital image processing	Lecture + examples	AI-based workflow visualization	Explain the stages of image processing
2	4	Components of an image processing system	Lecture + demo	AI-supported system mapping	Identify the components of an image processing system
	5	Elements of visual perception	Lecture + discussion	AI-assisted perception examples	Understand the role of human vision in image processing
	6	Light and the electromagnetic spectrum	Lecture + examples	AI-generated spectrum illustrations	Explain imaging in relation to light and spectrum
3	7	Image sensing and acquisition	Lecture + demo	AI-assisted acquisition examples	Understand image acquisition methods
	8	Image sampling and quantization	Lecture + problem solving	AI-generated digitization examples	Apply sampling and quantization concepts
	9	Basic relationships between pixels	Lecture + examples	AI-assisted pixel relation visualization	Understand adjacency, connectivity, and pixel relationships
4	10	Basic intensity transformation functions	Lecture + coding demo	AI-generated intensity transformation code	Apply basic intensity transformations
	11	Histogram processing	Lecture + demo	AI-assisted histogram interpretation	Analyze histogram-based image enhancement
	12	Fundamentals of spatial	Lecture +	AI-supported filtering	Understand spatial

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		filtering	discussion	examples	filtering basics
5	13	Smoothing low-pass spatial filters	Lecture + demo	AI-assisted denoising suggestions	Apply smoothing filters
	14	Sharpening high-pass spatial filters	Lecture + coding demo	AI-generated sharpening comparisons	Apply sharpening filters
	15	High-pass, band-reject, and band-pass filters	Lecture + examples	AI-supported filter construction	Understand advanced filter design
6	16	Combining spatial enhancement methods	Lecture + activity	AI-assisted enhancement planning	Integrate multiple spatial enhancement methods
	17	Background of frequency domain filtering	Lecture + discussion	AI-based concept explanation	Understand the background of frequency analysis
	18	Sampling and Fourier transform of sampled functions	Lecture + problem solving	AI-supported Fourier examples	Relate sampling to Fourier analysis
7	19	Discrete Fourier Transform of one variable	Lecture + examples	AI-generated transform demonstrations	Understand 1-D DFT basics
	20	Extensions to functions of two variables	Lecture + examples	AI-assisted 2-D transform visualization	Apply 2-D transform concepts
	21	Properties of 2-D DFT and IDFT	Lecture + discussion	AI-supported mathematical interpretation	Explain important transform properties
8	22	Basics of filtering in the frequency domain	Lecture + demo	AI-assisted frequency filtering guidance	Understand filtering in frequency domain
	23	Image smoothing using low-pass frequency filters	Lecture + coding demo	AI-generated smoothing workflows	Apply low-pass filtering in frequency domain
	24	Image sharpening using high-pass filters	Lecture + demo	AI-assisted sharpening comparisons	Apply high-pass frequency filtering
9	25	Selective filtering	Lecture + examples	AI-supported selective filter design	Understand selective filtering methods
	26	Introduction to morphological image processing	Lecture + discussion	AI-assisted morphology visualization	Explain the purpose of morphology
	27	Erosion and dilation	Lecture + demo	AI-generated morphological	Apply basic morphological operations

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
				examples	
10	28	Opening and closing	Lecture + coding demo	AI-supported shape refinement examples	Understand opening and closing operations
	29	Hit-or-miss transform	Lecture + examples	AI-assisted pattern matching support	Explain the hit-or-miss transform
	30	Basic morphological algorithms and reconstruction	Lecture + demo	AI-generated reconstruction examples	Apply morphological reconstruction concepts
11	31	Morphological operations on binary images	Lecture + discussion	AI-supported binary analysis	Understand binary morphology techniques
	32	Grayscale morphology	Lecture + demo	AI-assisted grayscale operation examples	Apply grayscale morphological processing
	33	Fundamentals of image segmentation	Lecture + examples	AI-based segmentation overview	Understand the role of segmentation
12	34	Point, line, and edge detection	Lecture + coding demo	AI-assisted edge analysis	Apply edge-based detection methods
	35	Thresholding techniques	Lecture + examples	AI-generated threshold selection suggestions	Understand threshold-based segmentation
	36	Region growing and splitting/merging	Lecture + demo	AI-assisted region analysis	Apply region-based segmentation methods
13	37	Clustering and superpixel segmentation	Lecture + examples	AI-supported clustering workflows	Understand clustering-based segmentation
	38	Graph cuts for image segmentation	Lecture + discussion	AI-generated graph-cut interpretation	Explain graph-cut segmentation
	39	Morphological watershed segmentation	Lecture + demo	AI-assisted watershed visualization	Apply watershed segmentation
14	40	Applications of digital image enhancement and segmentation	Lecture + case study	AI-based application mapping	Relate course concepts to real-world applications
	41	Recent trends in image processing and AI	Lecture + discussion	AI-supported literature exploration	Identify modern trends in image processing
	42	Course review and mini project discussion	Discussion	AI-assisted summary tools	Consolidate overall course understanding
15	43	Case Study of Unit I and II	Discussion	AI-supported case interpretation	Review image fundamentals and enhancement
	44	Case Study of Unit III and	Discussion	AI-assisted transform	Review frequency and

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		IV		and morphology review	morphology concepts
	45	Case Stu dy of Unit V	Discussion	AI-based segmentation	Consolidate segmentation

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

CLINICAL RESEARCH & ETHICS

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

SEMESTER – IV

Subject Code	:	25SM2408	**Credits**	:	2
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P-J	:	2-0-0-0			

Course Learning Objectives

This course enables students to:

- Understand the fundamentals of clinical research, its phases, and regulatory landscape.
- Gain familiarity with Good Clinical Practices (GCP), ethical frameworks, and guidelines in India and globally.
- Explore the relationship between clinical trials and innovation in medical technologies.
- Develop an understanding of protocol design, informed consent, and data integrity.
- Prepare for participation in design of clinical trials involving biomedical devices, software, and AI.

Teaching-Learning Process (General Instructions)

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

- Interactive lectures with real-world case discussions.
- Guest talks by clinical researchers and ethics committee members.
- Journal club-style analysis of landmark trials or device trials.
- Project-based learning: student teams will create mock clinical trial protocols.
- Exposure to basic ethics forms and regulatory documents (ICMR, CDSCO, DCGI).

UNIT – I	**06Hours**

Introduction to Clinical Research

Definition and scope. History and evolution of clinical research. Types: observational vs. interventional. Phases of clinical trials

UNIT – II	**06Hours**

Ethics in Clinical Research

Principles of biomedical ethics: autonomy, beneficence, non-maleficence, and justice. Nuremberg Code, Declaration of Helsinki, Belmont Report. Indian guidelines: ICMR Guidelines, New Drugs and Clinical Trials Rules. Ethics Committees: Composition and function.

UNIT – III	**06Hours**

Regulatory Framework

Roles of DCGI, CDSCO, ICMR, US FDA, EMA. IND, NDA, ANDA processes. Device trials: Software as a Medical Device (SaMD). Clinical Trial Registry of India (CTRI).

UNIT – IV	**06 Hours**

Trial Design and Documentation

Protocol writing. Inclusion and exclusion criteria. Case Report Forms (CRFs). Informed Consent Process. Adverse Event Reporting.

UNIT – V	**06 Hours**

Clinical Research in Medical Engineering

Device testing: preclinical and clinical stages. Integration of AI/ML in clinical research.
Data privacy and data management (GDPR, HIPAA basics). Real-world evidence (RWE) and digital biomarkers.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Explain the fundamentals and scope of clinical research	L2
CO2	Describe ethical principles, roles of ethics committees, and national guidelines	L2
CO3	Identify and compare global regulatory frameworks governing clinical trials	L3
CO4	Design basic components of a clinical trial protocol, including consent and data reporting	L4
CO5	Integrate principles of clinical research into medical engineering innovations	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	12	1	2	3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-
CO2	2	3	2	2	-	3	3	2	-	3	3	-	3	2	2
CO3	3	3	2	3	2	2	2	2	-	3	2	-	2	3	3
CO4	3	2	3	2	2	-	3	2	-	2	2	-	3	3	3
CO5	2	3	3	3	3	2	2	3	2	3	3	2	3	3	3

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

TEXTBOOKS:

- *Principles and Practice of Clinical Research* – John I. Gallin
- *Clinical Trials: A Methodologic Perspective* – Steven Piantadosi

REFERENCE BOOKS:

- *Good Clinical Practice Guidelines* – ICH E6
- ICMR National Ethical Guidelines for Biomedical and Health Research, 2017
- CDSCO Regulatory Guidelines

SUGGESTED APPLIED ACTIVITIES (TUTORIALS)

- Case studies on trial ethics violations
- Mock ethics committee role play
- Drafting an informed consent form for a wearable device trial
- Review and critique of a real protocol from CTRI
- Discussion on AI/ML bias and patient data handling

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

INTRODUCTION TO FILM STUDIES

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

SEMESTER – IV

Course Code	:	25SM2409	**Credits**	:	02
Hours /Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P-J	:	2–0–0-0			

Course Learning Objectives:

This Course will enable students to:

1. **Acquire** fundamental knowledge of the principles of critical analysis of film as a text (With the overview of Indian cinema after Independence)
2. **Develop** analytical and interpretative minds, and transform them into spoken and written modes of expression i.e., Debates, group discussions, writing, reviews, and scripts
3. **Apply** the writing strategies for writing film reviews.

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*** means it includes not only traditional lecture method, but different *type 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.

Module – I	**15 Hours**

History and Scope of Indian Cinema- the Beginnings (1896-1912)- The Silent Era (1913-1930), Growth of the Industry (1931-1947)-Post-Independence- Indian Cinema-Film aesthetics: Realism, Authorship, Language of the Film (Film, Language & Montage), Narrative Comprehension – Representations and Debates -Film and other arts-Career Prospects in Indian Cinema.

Module – II	**15 Hours**

Basic Frame- Shot-Scene-Sequence-Visual Grammar- Pioneers of film, Silent to audio era- The language of Cinema-Camera- Cinematography – Important functions and terms of Cinematography. –Science and Art of Lighting-Different Lightings-Cinematographic tools- Editing- What is a script? – Thinking process- Theatre Directions-Creating and Developing a script- Adapting a script- Character spin-off- Creating a sequel- Characterization-Qualities of a scriptwriter- Themes and social Obligations- Hands-on Experience.

TEXT BOOKS:

1. Saran, Renu. “History of Indian Cinema”. Diamond Books, 2012. Print.
2. Valicha, Kishore. “The Moving Image: A study of Indian Cinema”. Orient Longman Limited, 1988. Print.
3. Vasudevan, Ravi. “The Melodramatic Public: Film form and Spectatorship in Indian Cinema”. Orient Black Swan, 2012. Print.
4. Jain, Manju. “Narratives of Indian cinema”. Primus Books, 2009. Print.

REFERENCE BOOKS:

1. Dwyer, Rachel. “Cinema India: The Visual Culture of Hindi film”. Rutgers University Press, 2002. Print.
2. Corrigan, Timothy. “A Short Guide to Writing About Film”. 9th Edition, Pearson Global Edition. Print.

3. Joseph M. Boggs and Dennis W. Petrie. "The Art of Watching Films". Boston: McGraw-Hill, 2008. Print.
4. Richard, Barsam. "Looking at movies: An introduction to film". New York: W.W. Norton & Co., 2007. Print.
5. James, Monaco. "How to read a film: Movies, Media and Multimedia: Language, History, Theory". New York: Oxford University Press, 2000. Print.

Course Outcome	Description
At the end of the course the student will be able to:	
1	**Acquire** fundamental knowledge of the principles of critical analysis of film as a text (With the overview of Indian cinema after Independence)
2	**Develop** analytical and interpretative minds, and translate them into spoken and written modes of expression i.e. Debates, group discussions, writing, reviews, and scripts
3	**Develop** communicative competence – Competence in speaking and writing from different domains and **applying** an interdisciplinary approach in perception and expression.

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

CYBER CRIME, POLICIES, LAWS

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

SEMESTER – IV

Course Code	:	25SM2410	**Credits**	:	02
Hours /Week	:	02 Hour	**Total Hours**	:	30 Hours
L–T–P–J	:	2–0–0–0			

Course Learning Objectives:

This Course will enable students to:

1. Understand Cybercrime Fundamentals.
2. Explore Cybercrime Investigation Techniques.
3. Discuss Cybersecurity Incident Response.

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*** means it includes not only traditional lecture method, but different *type 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 analyse information rather than simply recall it.
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: Introduction to Cyber Security	**08 Hours**
Basic Cyber Security fundamentals, Types of Cyber Attacks, The 7 layers of Cyber Security, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, types of active attacks, passive Attacks, Software attacks, hardware attacks, Cyber Threats-, Cyber Crime, Cyber terrorism, Cyber Espionage.	
UNIT – II: Cyberspace and the Law & Cyber Forensics	**07 Hours**
Introduction, Cyber Security Regulations, Roles of International Law, Cyber forensics, Digital forensics lifecycle: Types of forensics investigation, challenges in computer forensics.	
UNIT – III: Cybercrime: Mobile and Wireless Devices	**07 Hours**
Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era.	
UNIT – IV: Cyber Security: Organizational Implications	**08 Hours**
Drama within liberal studies serves as a vehicle for social commentary and advocacy. By exploring social issues such as discrimination, inequality, and justice, drama becomes a powerful tool for initiating conversations and inspiring action. Through the examination of theatrical texts and the creation of original performances, students develop the skills to articulate their perspectives and communicate their ideas effectively. This process empowers them to become advocates for change and contribute to a more inclusive and just society.	

Course Outcome	Description
At the end of the course the student will be able to:	
1	Analyse cyber-attacks, types of cybercrimes, cyber laws and also how to protect them self and ultimately the entire Internet community from such attacks.
2	Interpret and forensically investigate security incidents.
3	Apply policies and procedures to manage Privacy issues.

TEXT BOOKS:

1. Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes,Computer Forensics and Legal Perspectives, Wiley
2. B.B.Gupta,D.P. Agrawal, Haoxiang Wang, Computer and CyberSecurity: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335,2018.

REFERENCE BOOKS:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&FGroup.

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

YOGA AND MEDITATION

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

SEMESTER – IV

Course Code	:	25SM2411	**Credits**	:	02
Hours/Week	:	02 Hour	**Total Hours**	:	30 Hours
L–T–P–J	:	2–0–0–0			

Course Learning Objectives:

This Course will enable students to:

1. Promoting positive health and holistic wellness.
2. Imparting skills to introduce Yoga awareness for health among general public.
3. To enable students to become competent and committed professionals willing to perform as Yoga performer.
4. To make student to use competencies and skills needed for becoming effective Yoga individual.
5. To enable student to understand the type of Yoga.

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*** means it includes not only traditional lecture method, but different *type 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. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I: Introduction to Yoga and Yogic Practices **06 Hours**

Yoga: Etymology, definitions, aim, objectives and misconceptions, Yoga: Its Origin, history and development, Rules and regulations to be followed by Yoga Practitioners, Introduction to major schools of Yoga (Jnana, Bhakti, Karma, Patanjali, Hatha), Introduction to Yoga practices, Introduction to Surya Namaskar.

UNIT – II: Foundation of Yoga and Meditation **06 Hours**

Yoga, Diet and Nutrition, Practical – I (Yogasana and Meditation I), Practical II (Shatkarma, Pranayam and Meditation II)

UNIT – III: Types & techniques of Meditation II **06 Hours**

Meditation: Meditation- Introduction, definition, Concentration and meditation, OM Meditation.
Types of Meditation: Tantra: Yantra and Mantra for meditation, Japa (chanting) meditation, AjapaJapa Meditation, Shoonya Meditation, Yoga Nidra.

UNIT – IV: Relationship between Meditation & Yoga (Mental Health Aspects) **06 Hours**

Yoga, Diet and Nutrition, Practical – I (Yogasana and Meditation I), Practical II (Shakira, Pranayama and Meditation II)

Yoga, Diet and Nutrition, Practical – I (Yogasana and Meditation I), Practical II (Shatkarma, Pranayam and Meditation II). Yoga and Mental Health, Define mental health, Holistic health, Medical & Yogic perspective

Yoga and Mental Health, Define mental health, Holistic health, Medical & Yogic,,perspective, Mental Hygiene, Mental Hygiene and roll of yoga in mental, hygiene, Relationship between mind and body, Meditation – Yoga, Swara yoga – the balance of life, The mind and personality

Yoga & Stress Management: Human Psyche: Yogic and modern concept, Behaviour and consciousness frustration, Conflicts, Concept of stress according to modern science and yoga, Stress and Personality, Role of yoga in life management Mental Hygiene and roll of yoga in mental hygiene, Relationship between mind and body. **Meditation – Yoga,** Swara yoga – the balance of life, The mind and personality

Yoga & Stress Management: Human Psyche: Yogic and modern concept, Behaviour and consciousness frustration, Conflicts, Concept of stress according to modern science and yoga, Stress and Personality, Role of yoga in life management

UNIT – V: Yogasana, Pranayam and Meditation Practical's **06 Hours**

Yogasanas

Standing Asana: Tadasana, Trikonasana, Urdhahastotanasana, Vrikshasana, Ardhchakrasna, Padhastanasana, Ashwasthasana.

Sitting Asana: Padmasana, Vakrasana, Ardhamatsyendrasana, Janusirsasana, Paschimottanasana, Vajrasana, Ushtrasana, Shasankasana, Gomukhasana, Mandukasana, Bhadrasana, Singhasana.

Prone Lying Asana: Bhujangasana, Shalabhasana, Dhanurasana, Makarasana

Supine Lying Asana: Pawanmuktasana and its variation, setubandhasana, sarvangasana, Ardhahalasana, Uttanpadasana, Halasana, Naukasana , Cakrasana , Markatasana, Shavasana.

Surya Namaskar, Pranayama: Breath Awareness. Yogic Breathing. Nadishodhan Pranayama. Suryabhedi. Ujjayi. Shitali. Sitkari. Bhastrika. Bhramari.

Bandha and Mudra: JalandharaBandha, UddiyanaBandha, MulaBandha, Tri Bandha. Yoga Mudra, Shanmukhi Mudra, shambhavi mudra, VipareetKarni Mudra.

Course Outcome	Description
At the end of the course the student will be able to:	
1	Development of relaxation techniques.
2	Improved concentration and cognitive function.
3	Promotion of overall health and well-being.

Text Book and Reference Book:

1. Concentration and Meditation by swami Sivananda Saraswati
2. Yoga and Kriya by Swami Satyananda Saraswati
3. Yoga &Mental Health by R. S. Bhogal
4. Yoga & Modern Psychology by KaivalyadhamAsharam
5. Yoga for Stress Management by Sri Venkatkrishnan
6. Yoga for Stress Relief by Swami Shivapramananda
7. Yoga Nidra by Swami Styananda Saraswati
8. Certification of Yoga Professionals – Official Guide book

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

DESIGN STUDIO: AI-ENABLED HEALTHCARE SYSTEMS

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

Course Code:25SM2406	**Course Type:** Program Core Course
Programs Offered: CSME	**Total Hours:** 30 Hours
Semester: IV	**Hours/Week:** 02 Hours
Credits: 02	**L – T – P – J:** 1 – 1 –0 – 0
Prerequisites: Programming Fundamentals (Python)	

Course Vision

This course develops the ability to design human-centered intelligent solutions by understanding the Design Thinking framework and rapid AI prototyping, empowering students to build empathetic and effective software systems.

Course Objectives

At the end of course, students will able to:

1. Understand the fundamental stages of the Design Thinking framework.
2. Identify and frame real-world problems suitable for artificial intelligence interventions.
3. Design intelligent interactions and map user journeys for AI-enabled systems.
4. Implement functional prototypes using rapid prototyping tools and AI APIs.
5. Evaluate prototype reliability, usability, and user trust through continuous testing.

Course Outcomes (CO)

At the end of the course the student will be able to:

CO	Course Outcomes	Bloom taxonomy level
CO1	Understand the principles of Design Thinking and their application to AI systems.	L3
CO2	Formulate user-centric problem statements and "How Might We" questions.	L3
CO3	Design interfaces and user journeys that account for AI-specific interaction challenges.	L3
CO4	Develop high-fidelity intelligent prototypes using modern Python frameworks and APIs.	L3
CO5	Evaluate the usability and performance of AI applications through structured user testing.	L2& L3

Broad Course Description

It is a foundational course bridging user-centric problem-solving and artificial intelligence. It combines the iterative Design Thinking framework with hands-on Intelligent System Prototyping. The curriculum emphasizes experiential learning through a weekly structure consisting of a 1-hour theoretical framing lecture followed by a 2-hour rigorous, project-based development task. Students will learn to empathize with users, ideate AI-driven solutions, and build functional prototypes using modern AI APIs (e.g., Hugging Face, OpenAI) and deployment frameworks (e.g., Streamlit, Gradio). By the end of the course, students will be able to translate complex user needs into deployable, intelligent software applications.

Detailed Course Syllabus

This course follows a project-based learning model where each week consists of a 1-hour lecture introducing concepts followed by a 2-hour hands-on project task. Students build a single integrated project across all 15 weeks, with each week adding a new layer of functionality and design.

Phase 1 — Foundations & AI Literacy (Weeks 1–3)
Authentic Intelligence Layer — *Authentic Intelligence Layer: Before building, learn to question. Students examine how AI tools are trained, what biases they may carry, and where they fail. Every tool evaluation must include a documented limitation and at least one edge case where the tool gives a wrong or misleading output.*

Week 1: Introduction to AI-Integrated Design Thinking	
LECTURE (1 HR) Design thinking lifecycle (Empathize, Define, Ideate, Prototype, Test). Role of AI as a co-designer. Overview of modern AI tools: ChatGPT, Midjourney, GitHub Copilot, Figma AI. Ethics and responsible AI design.	**PROJECT TASK (2 HRS)** Explore 5 AI tools. Document their capabilities, limitations, and potential use-cases in a 1-page 'AI Tool Map'. Share findings with peers in a discussion session. *AI Authenticity Check: For each tool explored, answer — Who built it? On what data was it trained? What groups might be underrepresented? What happens when you push it to its edge? Document at least one hallucinated or biased output.*
Week 2: Prompt Engineering for Designers & Developers	
LECTURE (1 HR) Anatomy of a good prompt. Prompt patterns: zero-shot, few-shot, chain-of-thought. Prompting for text, image, and code generation. Iterative prompt refinement strategies.	**PROJECT TASK (2 HRS)** Design 10 prompts across 3 different problems (UI generation, code snippet, data summary). Compare AI outputs, iterate, and submit a 'Prompt Cookbook' with annotated results. *Reliability Snapshot: Run each prompt 3 times and compare outputs. Note variance. Flag any prompt where >1 run produced inconsistent or incorrect results. Conclude: which tasks should NOT be delegated to AI without human verification?*
Week 3: AI-Assisted Ideation & Brainstorming	
LECTURE (1 HR) SCAMPER technique combined with AI assistance. How LLMs aid divergent thinking. Concept clustering and affinity diagrams. Tools: Miro, FigJam, Notion AI.	**PROJECT TASK (2 HRS)** Pick a real-world problem. Use an AI tool to generate 50+ ideas in 30 minutes. Cluster, evaluate, and select the top 3 concepts. Present a visual Idea Board. *Bias in Ideation: After clustering, ask — did the AI ideas skew toward a particular demographic, region, or solution type? Did it miss culturally specific solutions? Write a 3-sentence 'Bias Audit' on your Idea Board.*

Phase 2 — AI in UI/UX &Prototyping (Weeks 4–7)
Authentic Intelligence Layer — *Authentic Intelligence Layer: AI design tools optimise for aesthetics that reflect their training data. Students must actively audit generated designs for inclusivity, accessibility, and cultural appropriateness.*

Week 4: User Research with AI Assistance [Research]	
LECTURE (1 HR) User personas and empathy maps. AI-powered survey analysis. Sentiment analysis fundamentals. Tools: Typeform with GPT integration, Dovetail.	**PROJECT TASK (2 HRS)** Conduct a 5-person micro-survey on a campus problem. Feed responses to an AI to extract themes and insights. Build 2 data-driven user personas with empathy maps. *Empathy Audit: Where did the AI persona differ from your own reading? Did the AI flatten nuance or miss emotional context? Note 2 things the AI got wrong and how you corrected them. This is the human layer AI cannot replace.*
Week 5: Wireframing & Lo-Fi Prototyping with AI [Design]	
LECTURE (1 HR) Lo-fi vs hi-fi prototyping differences. Wireframing with Figma and Uizard AI. Translating text descriptions into UI wireframes. Feedback loops using AI review.	**PROJECT TASK (2 HRS)** Use Uizard or Figma AI to generate wireframes for the chosen concept from Week 3. Iterate based on a peer review session. Submit 5 annotated wireframe screens. *Inclusion Check: Does the AI-generated UI default to able-bodied, Western, or English-speaking assumptions? Test each screen for colour contrast (min 4.5:1), keyboard navigability, and screen-reader compatibility. Document fixes made.*
Week 6: Hi-Fi Design, Design Systems & AI Styling [Design]	
LECTURE (1 HR) Design systems and component libraries. Generating color palettes, typography, and icons with AI. Figma tokens and variables. Accessibility basics (WCAG 2.1).	**PROJECT TASK (2 HRS)** Build a mini design system for your project: primary colors, 2 font styles, 5 reusable components (button, card, input, navigation, badge). Use AI to suggest and refine choices. *Style Bias Test: Ask the AI to generate 3 different palette suggestions. Compare — do they reflect a global user base or a specific cultural aesthetic? Document one palette choice you changed based on your target users' cultural context*
Week 7: Interactive Prototyping & Usability Testing [Testing]	
LECTURE (1 HR) Clickable prototypes in Figma. Usability heuristics (Nielsen's 10 principles). Running a guerrilla usability test. Analyzing session recordings with AI assistance.	**PROJECT TASK (2 HRS)** Convert wireframes into a clickable Figma prototype. Run 3 usability tests with classmates. Use AI to summarize feedback. Document 5 key usability findings and propose fixes. *Feedback Fidelity: Did the AI summary miss sarcasm, frustration, or hesitation that you observed in person? Write a 'Translation Note' — 2–3 things the AI summary got wrong and how lived observation corrected the record.*

Phase 3 — AI-Powered Development (Weeks 8–12)
Authentic Intelligence Layer- AI writes code fast — but fast is not always correct, secure, or fair. Students must review every AI-generated function for correctness, security vulnerabilities, unintended bias, and explainability. No AI-generated code ships without a human sign-off checklist.

Week 8: AI-Assisted Frontend Development [Coding]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
GitHub Copilot and Cursor IDE walkthrough. Effective prompting for HTML/CSS/JS generation. Code review with AI. Writing accessible, responsive UI code.	Implement the landing page of your project using AI-assisted coding (Copilot/Cursor). Must be responsive and include: navigation bar, hero section, and one interactive component. *Code Review Checklist (mandatory for every AI-generated snippet): (1) Do I understand every line? (2) Is there a security risk? (3) Does it handle edge cases? (4) Is it accessible? Attach the completed checklist to your submission.*

Week 9: Backend Logic & API Integration with AI [Coding]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
RESTful API fundamentals. Using AI to generate CRUD endpoints. Integration patterns using fetch and axios. Introduction to serverless functions.	Build a simple backend (Node.js/Express or Python Flask) for your project with at least 3 API endpoints. Use AI for boilerplate generation and test case writing. Document with Postman. *Security Audit: Run your AI-generated endpoints through a basic OWASP checklist. Did Copilot/ChatGPT introduce any SQL injection risks, hardcoded secrets, or missing input validation? Document what you found and how you fixed it.*

Week 10: Integrating AI Models into Applications [AI Dev]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
OpenAI API and Hugging Face platform overview. Embedding AI features: chatbots, image recognition, text summarization. API key management and cost awareness.	Add one AI-powered feature to your project using an external AI API (e.g., chatbot, auto-summarizer, or image tagger). Document the integration process and edge cases. *Hallucination Analysis: Deliberately prompt your integrated AI feature with ambiguous, edge-case, or culturally specific inputs. Record at least 3 cases where it produced a confident but incorrect or inappropriate response. Reliability Evaluation: Score your feature on a 1–5 scale for accuracy, consistency, and failure transparency. Bias Testing: Test with inputs from at least 3 demographic or linguistic contexts. Does the output quality differ? Document findings and mitigations implemented.*

Week 11: Data, Databases & AI-Driven Insights [Data]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
SQL vs NoSQL comparison. Connecting a database to your application. Using AI for query generation and optimization. Basic data visualization with Chart.js and D3.	Connect a database (SQLite or Firebase) to your project. Use AI to generate queries. Display at least one data visualization on a dashboard page of your app. *Data Representation Audit: Who is in your dataset? Who is missing? If you use a public dataset, check its documentation for known biases. If your dataset is self-collected, identify which groups may be underrepresented. Include a 'Data Limitation' note in your dashboard.*

Week 12: Testing, Debugging & Code Quality with AI [QA]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
Unit testing fundamentals with Jest and Pytest. AI-assisted debugging workflows. Code linting and formatting. Writing readable inline documentation.	Write unit tests for at least 5 functions in your project using AI to generate test stubs. Fix 3 AI-identified bugs. Add inline documentation to all key functions. *Explainability: For your AI feature, add a logging layer that records — what input was given, what the AI returned, and what confidence score (if available) was attached. If no confidence score exists, note this as a design risk. Trust Score: Assign a trust score (Low / Medium / High) to your AI feature based on your Week 10 test results. Justify the rating in 2 sentences. Safe Fallback: Implement a fallback path — what does your app do when the AI API is down, returns an error, or returns a low-confidence answer?*

Phase 4 — Integration, Launch &Presentation (Weeks 13–15)
Authentic Intelligence Layer: Deployment is a moral act. Before any system goes live, students must conduct a structured ethical review — not as a box to tick, but as a genuine gate. The question 'Should this be deployed?' is mandatory, not optional. A technically working system is not automatically a responsible one.

Week 13: Full-Stack Integration & Deployment [DevOps]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
Connecting frontend, backend, and database layers. Environment variables and secrets management. Deployment to Vercel, Render, and Railway. CI/CD basics with GitHub Actions.	Deploy your complete project to a live URL. Ensure all three layers work end-to-end. Submit a deployment checklist and a working demo link for peer review. *Pre-Deployment Ethical Review (mandatory before going live): (1) Who could be harmed by this system if it fails? (2) Is user data handled responsibly and transparently? (3) Does the system behave equitably across different user groups? (4) Is there a mechanism for users to report errors or seek human help? Complete and submit the review alongside your deployment checklist.*

Week 14: Project Refinement, Storytelling & Portfolio [Presentation]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
Crafting a compelling design and tech narrative. Case study structure and writing. Portfolio platforms: Notion, Behance, GitHub. Writing impactful README files.	Create a case study covering: problem, research, design decisions, tech stack, AI tools used, challenges, and outcomes. Publish it on Notion or Behance for public access. *AI Honesty Section (mandatory in case study): List every AI tool used and its specific role. Name at least 2 instances where AI produced an incorrect or unhelpful output and describe how you identified and fixed it. State clearly: what this project could NOT have done without human judgment.*

Week 15: Final Project Demo Day & Peer Review [Showcase]

LECTURE (1 HR)	PROJECT TASK (2 HRS)
Demo day best practices. Giving and receiving constructive critique. Evaluating AI ethics in your own project. Reflection and growth mindset frameworks.	Present complete project in a 10-minute demo (5 min presentation + 5 min Q&A). Submit a final reflection (400 words) on: what AI changed in your process, key learnings, and what you would do differently.

MANDATORY CLOSING QUESTION — Should This System Be Deployed? Why?

Every student must answer this question in their presentation and written reflection. Acceptable answers include 'Yes, with the following conditions...', 'Not yet, because...', or 'No, because...' — all are valid if well-argued. The answer is graded on reasoning quality, not on whether the system is deployed. Evaluation criteria: (1) Have you identified all significant failure modes? (2) Have you assessed who benefits and who bears the risk? (3) Is there a monitoring and correction plan if the system harms a user? (4) Would you be comfortable if the people most affected by this system could see exactly how it works?

Laboratory Component (AI Integrated) (15-Week Project Task Schedule)

Week 1: Team formation and broad problem domain selection.
Week 2: Conduct mock user interviews and draft user personas.
Week 3: Frame the formal problem statement and generate "How Might We" questions.
Week 4: Brainstorm AI-driven solutions and select the primary concept to pursue.
Week 5: Create a step-by-step visual storyboard or user journey map.
Week 6: Build a low-fidelity clickable wireframe (using Figma or paper) of the system.
Week 7: Write basic Python scripts to ping an AI API endpoint and retrieve outputs.
Week 8: Build a basic command-line conversational prototype using prompt engineering.
Week 9: Implement a pre-trained computer vision or classification model script.
Week 10: Build an interactive web app interface using Streamlit or Gradio.
Week 11: System Integration: Connect the AI backend logic to the web frontend.
Week 12: Refine the UI to include AI confidence scores, explanations, and user fallbacks.
Week 13: Conduct internal testing for edge cases, failures, and system constraints.
Week 14: Conduct live usability tests with target users and measure task success.
Week 15: Final Project Exhibition: 10-minute pitch and live prototype demonstration.

DLI: Building LLM Applications with Prompt Engineering

An AI Factory is NVIDIA's concept for a full-stack, production-ready GPU-accelerated platform that integrates compute infrastructure, AI software (NeMo, NIM), data pipelines, and governance into a repeatable operating model for building AI at scale.

CO – PO / PSO Mapping

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	3
CO1	3	2	1	-	-	1	-	-	-	-	-	-	-	2
CO2	1	3	2	-	-	2	-	-	2	-	-	-	-	3
CO3	1	2	3	1	2	-	-	-	-	-	-	-	-	3
CO4	2	2	3	-	3	-	-	-	-	-	-	-	-	2
CO5	-	2	2	3	2	2	-	2	3	3	-	-	-	3

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

Assessment Scheme (100% CIA) -Integrated Evaluation (Theory + Practical)

Component	Description	Weightage
Weekly Task Submissions	15 individual deliverables (Weeks 1–14)	40%
Mid-Term Review	Prototype + design system presentation (Week 7)	20%
Final Project Demo	Live deployed product demonstration (Week 15)	25%
Case Study & Reflection	Published case study + reflection essay	15%

Tools & Technology Stack

Category	Tools
AI & LLMs	ChatGPT / OpenAI API, Gemini, Hugging Face, Claude, NVIDIA LLM NIM
Design & Prototyping	Figma, Uizard AI, Miro, FigJam, Adobe XD
Development (AI-assisted)	GitHub Copilot, Cursor IDE, VS Code
Frontend	HTML5, CSS3, JavaScript, React (optional)
Backend	Node.js + Express, Python + Flask
Database	SQLite, Firebase, MongoDB Atlas
Deployment	Vercel, Render, Railway, GitHub Pages
Version Control	Git + GitHub (CI/CD via GitHub Actions)
Documentation	Notion, Postman, README.md, Behance

Primary Textbooks

• Don Norman – The Design of Everyday Things
• Saleema Amershi et al. – Guidelines for Human-AI Interaction (Microsoft)

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

MATLAB PROGRAMMING

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

SEMESTER – IV

Subject Code	:		**Credits**	:	02
Hours / Week	:	02 Hours	**Total Hours**	:	30 Hours
L–T–P–S	:	1–1–0–0			

Course Learning Objectives:

This Course will enable students to:

1. Acquire proficiency in core MATLAB functionality, encompassing data analysis, visualization, modeling, and programming. By **implementing** a standardized data analysis workflow, students will develop the skills necessary to tackle diverse challenges across science and engineering domains effectively.
2. Gain knowledge in fundamentals of MATLAB, commonly used features and workflows. Through hands-on practice, they will become adept at navigating the MATLAB environment and utilizing the MATLAB language for **analysing** scientific and engineering data effectively.

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*** means it includes not only traditional lecture method, but different *type 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 analyse information rather than simply recall it.
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	**06Hours**

INTRODUCTION:

Using the MATLAB Desktop, Desktop Overview, Getting Data into MATLAB, Customizing the MATLAB Environment, Creating Informative Scripts

CREATING AND MANIPULATING ARRAYS-Creating Arrays, Concatenating Arrays, Reshaping Arrays

ACCESSING DATA IN ARRAYS-Extract subsets of arrays, and modify elements in an array.

MATHEMATICAL AND STATISTICAL OPERATIONS WITH ARRAYS-Performing Operations on Arrays, Calculating Statistics of Vectors, Using Statistical Operations on Matrices, Matrix Multiplication

(Text Book-1: Chapter 1)

UNIT – II	**06 Hours**

Visualizing Data in 2D and 3D-Identifying Available Vector Plot Types, Customized Annotations, Customizing Plot Properties, Axis Control, Plotting Multiple Columns, Visualizing Matrices, Exporting Figures

Conditional Data Selection-Logical Operations and Variables, Counting Elements, Logical Indexing

Review Project I - Project - Halfway Review of Fundamentals

TABLES OF DATA-Storing Data in a Table, Sorting Table Data, Extracting Portions of a Table, Extracting Data from a Table, Exporting Tables

ORGANIZING TABULAR DATA-Course Example - Premier League Team Information, Combining Tables, Table Properties, Indexing into Cell Arrays ***(Text Book-1: Chapter 2)***	
UNIT – III	**06 Hours**
SPECIALIZED DATA TYPES Working, operating on Dates and Times, Representing Discrete Categories **Preprocessing Data-**Course Example - Preprocessing Electricity Usage Data, Normalizing Data, Working with Missing Data, Interpolating Missing Data, **Common Data Analysis Techniques-**Course Example - Analyzing Electricity Consumption, Smoothing Data, Linear Correlation, Polynomial Fitting **Programming Constructs-**Course Example - Comparing Prices, User Interaction, Decision Branching, For Loops, While Loops **Increasing Automation with Functions** Creating and Calling Functions, Function Files, Workspaces, MATLAB Path and Calling Precedence, **Troubleshooting Code-**Finding Syntax Errors, Inspecting Variable Values, Stepping Through Code, Debugging Functions **Review Project II** ***(Text Book-1: Chapter 3)***	
UNIT – IV	**06 Hours**
MATLABONRAMP **Commands** Enter Commands, Name Variables, Save and Load Variables, Use Built-in Functions and Constants, **MATLAB Desktop and Editor**- MATLAB Desktop and Editor, The MATLAB Editor, Run Scripts, Debug MATLAB Code ,**Vectors and Matrices**- Manually Enter Arrays, Create Evenly Spaced Vectors, Create Arrays with Functions, **Array Indexing and Modification**–Indexing, Index into Arrays, Extract Multiple Elements Change Values in Arrays, **Array Calculations** -Perform Array Operations on Vectors, **Function Calls** - Request Multiple Outputs in Function Calls, **Documentation** - MATLAB Documentation, Use MATLAB Documentation, **Plots -** Plot Vectors, Annotate Plots **Data Import-** Import Tool, Import Data as a Table, **Logical Arrays** - Logical Indexing **Programming** - Programming Constructs, Decision Branching, For Loops, **Final Project** - Stellar Motion, Compare Stellar Spectra ***(Text Book-1: Chapter 4)***	
UNIT – V	**06 Hours**
Image Processing Onramp - image processing techniques - Images in MATLAB, Image Segmentation, Pre-processing and Postprocessing Techniques, Classification and Batch Processing **Image Processing with MATLAB** -Working with Image Data, Pre-processing, Colour Segmentation, Texture Segmentation, Improving Segmentations, Finding and Analysing Objects, Detecting Edges and Shapes, Batch Processing, Aligning Images with Image Registration ***(Text Book-1: Chapter 5)***	

Course Outcome	Description	Bloom's Taxonomy Level
1	**Demonstrate** proficiency in navigating MATLAB's environment, customizing scripts, and executing array manipulation techniques (create,	**L2**

	concatenate, reshape) alongside efficient execution of mathematical and statistical operations on arrays.	
2	**Apply** visualization techniques effectively in 2D and 3D, customize plots, and control properties. Acquire proficiency in conditional data manipulation, logical operations, indexing, and efficient organization of tabular data.	**L3**
3	**Apply** preprocessing techniques such as normalization, handling missing data, and interpolation, and analyze data using common techniques like smoothing, correlation analysis, and polynomial fitting.	**L3**
4	**Develop** MATLAB programming proficiency, employing decision branching, loops, and function creation. Enhance troubleshooting skills by identifying syntax errors, inspecting variable values, and debugging functions, while increasing automation and code efficiency.	**L3**
5	**Apply** acquired knowledge to a final project, comparing stellar spectra, demonstrating mastery of MATLAB fundamentals in real-world contexts using advanced plotting, data import and practical image processing workflows.	**L3**

TEXT BOOKS:

1. Holly Moore, "MATLAB for Engineers" 5th Edition, Pearson, 2017

REFERENCE BOOKS:

1. **Stephen J. Chapman,"** MATLAB Programming for Engineers",4th edition, Thomson, 2008

E-Resources:

1. MATLAB Fundamentals | Self-Paced Online Courses - MATLAB & Simulink (mathworks.com)
2. Page not found | Self-Paced Online Courses - MATLAB & Simulink (mathworks.com)
3. Image Processing Onramp | Self-Paced Online Courses - MATLAB & Simulink (mathworks.com)
4. Image Processing with MATLAB | Self-Paced Online Courses - MATLAB & Simulink (mathworks.com)

Activity Based Learning (Suggested Activities in Class)

1. Hands-On MATLAB Desktop and Array Operations
2. Comprehensive Projects and Advanced Plotting

MATLAB PROGRAMMING LABROTARY

Following are experiments to be carried out using MATLAB programming language

1. Introduction & Features of MATLAB and Basic Command Syntax.
2. Writing simple programs to practice knowledge of variable, data types and Operators.
3. Writing different programs with the help of if-else, break, continue loops in MATLAB.
4. Writing different programs with the help of for, while loops in MATLAB.
5. Writing programs related with arrays, Matrices and vectors.
6. To create and run Script File in MATLAB with In-BUILD functions.
7. Writing programs related to visualize 2D and 3D plots & graphs using MATLAB.
8. Write a program perform image segmentation and classification using MATLAB.
9. Write a program to detect edges in images and perform batch processing using MATLAB

AI Integration & Authentic Intelligence Reflection (AI² Pedagogy)

• AI Integration:

Module–1: MATLAB Fundamentals and Array Operations

AI supports understanding of MATLAB environment, array manipulation, and mathematical operations through intelligent code suggestions, automated visualization, and debugging assistance. It enhances problem-solving in numerical computing and engineering data analysis.

Module–2: Data Visualization and Tabular Data Handling

AI facilitates intelligent visualization of engineering datasets, automated chart generation, logical indexing, and efficient table manipulation. It assists students in interpreting patterns, trends, and relationships in structured data.

Module–3: Specialized Data Types and Programming Constructs

AI enhances preprocessing, normalization, interpolation, and statistical analysis of real-world datasets. It supports automation of repetitive programming tasks, function generation, and debugging for efficient MATLAB programming.

Module–4: MATLAB Onramp and Automation

AI assists in learning MATLAB commands, scripts, array calculations, plotting, logical indexing, and debugging. It improves programming productivity through intelligent recommendations and automated workflow support.

Module–5: Image Processing with MATLAB

AI strengthens understanding of image segmentation, preprocessing, feature extraction, and object analysis using MATLAB image processing tools. It supports healthcare imaging, biomedical analysis, and intelligent computer vision applications.

Authentic Intelligence Reflection:

Module–1: MATLAB Fundamentals and Array Operations

Encourages students to critically analyze numerical computations, matrix operations, and engineering problem-solving using MATLAB programming.

Module–2: Data Visualization and Tabular Data Handling

Promotes analytical thinking in interpreting visual data representations and extracting meaningful insights from engineering datasets.

Module–3: Specialized Data Types and Programming Constructs

Develops authentic understanding of preprocessing, statistical analysis, and automation techniques for solving real-world engineering problems efficiently.

Module–4: MATLAB Onramp and Automation

Encourages students to design structured MATLAB programs, debug logical errors, and automate engineering computations systematically.

Module–5: Image Processing with MATLAB

Helps students understand practical implementation of image processing algorithms and critically evaluate segmentation and classification techniques in healthcare and engineering applications.

AI-Powered Assignments

Assignment 1: AI-Generated MATLAB Array and Matrix Operations

Prompt for Students

"Generate MATLAB code to perform array creation, reshaping, concatenation, matrix multiplication, and statistical analysis on engineering datasets."

Student Tasks

1. Analyze whether the generated MATLAB code correctly performs array operations and matrix computations.
2. Identify logical errors, dimension mismatches, and inefficient calculations.
3. Improve the code using optimized MATLAB built-in functions and vectorized operations.

Assignment 2: AI-Assisted Data Visualization and Table Analysis

Prompt for Students

"Generate MATLAB code to visualize engineering data using line plots, histograms, scatter plots, and heatmaps while storing data in tables and performing statistical analysis."

Student Tasks

1. Verify whether visualization techniques correctly represent the dataset.
2. Identify issues in table handling, indexing, and plotting methods.
3. Improve visualization quality using annotations, legends, customized axes, and logical indexing.

Assignment 3: AI-Based MATLAB Programming and Automation

Prompt for Students

"Generate MATLAB code for preprocessing electricity usage data including normalization, interpolation, smoothing, correlation analysis, and automation using functions."

Student Tasks

1. Analyze whether preprocessing and statistical analysis are correctly implemented.
2. Identify errors such as incorrect interpolation, inefficient loops, and missing-value handling problems.
3. Improve the program using modular functions, vectorization, and automated workflows.

Assignment 4: AI-Generated Image Processing with MATLAB

Prompt for Students

"Generate MATLAB code for medical image preprocessing, segmentation, edge detection, and object classification using image processing toolbox functions."

Student Tasks

1. Analyze whether preprocessing and segmentation techniques are correctly implemented.
2. Identify issues in thresholding, segmentation accuracy, and image enhancement methods.
3. Improve the algorithm using advanced preprocessing, morphological operations, and optimized feature extraction techniques.

Mini Project Components

Project 1: MATLAB-Based Engineering Data Analyzer
Problem Statement
Develop a MATLAB application for analyzing engineering datasets using matrix operations, statistical analysis, and data visualization techniques.
Features

- Data import and preprocessing
- Statistical analysis and visualization
- Correlation and trend analysis
- Automated reporting using MATLAB scripts

Project 2: Smart Electricity Consumption Analysis System
Problem Statement
Design a MATLAB-based system to preprocess, analyze, and visualize electricity consumption data for identifying usage trends and anomalies.
Features

- Missing value handling
- Data normalization and smoothing
- Correlation and forecasting analysis
- Automated graphical reports

Project 3: Medical Image Processing and Classification System
Problem Statement
Build a MATLAB application for preprocessing, segmentation, and classification of medical images.
Features

- Image enhancement and filtering
- Edge and object detection
- Segmentation and feature extraction
- Basic classification and evaluation

CO, PO and PSO Mapping Table

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	3
CO1	3	2	-	-	2	-	-	1	-	-	-	2	1	1
CO2	3	3	2	1	3	-	-	1	1	1	-	3	2	2
CO3	3	3	3	2	3	-	-	1	1	1	-	3	2	2
CO4	3	3	3	3	3	1	-	2	2	1	1	3	2	3
CO5	3	3	3	3	3	1	1	2	2	2	2	3	3	3
3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)														

Department of Computer Science & Medical Engineering

45 Hours Weekly Lecture Plan

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
1	1	Introduction to MATLAB Environment	Lecture + Demo	AI-assisted MATLAB navigation	Understand MATLAB desktop
	2	MATLAB Workspace and Scripts	Coding Demo	AI-generated scripts	Create MATLAB scripts
	3	Creating Arrays and Matrices	Practical Session	AI-assisted array creation	Perform array operations
2	4	Concatenation and Reshaping Arrays	Coding Demo	AI-generated reshaping examples	Manipulate arrays
	5	Accessing and Modifying Data	Lecture + Examples	AI-assisted indexing	Access array elements
	6	Mathematical Operations on Arrays	Problem Solving	AI-generated matrix operations	Apply matrix calculations
3	7	Statistical Operations in MATLAB	Lecture + Demo	AI-based statistical analysis	Compute statistics
	8	Matrix Multiplication	Coding Practice	AI-assisted multiplication steps	Perform matrix multiplication
	9	2D Data Visualization	Lecture + Demo	AI-generated plotting suggestions	Visualize engineering data
4	10	Plot Customization and Axis Control	Coding Demo	AI-assisted annotations	Customize plots
	11	Visualizing Matrices and Multiple Columns	Practical Session	AI-generated matrix visualization	Represent multidimensional data
	12	Exporting Figures and Reports	Lecture + Demo	AI-assisted export automation	Generate graphical reports
5	13	Logical Operations and Indexing	Lecture + Examples	AI-assisted logical analysis	Apply logical indexing
	14	Tables and Data Storage	Coding Demo	AI-generated table operations	Store tabular data
	15	Sorting and Extracting Table Data	Practical Session	AI-assisted table indexing	Manipulate tables
6	16	Combining Tables and Cell Arrays	Lecture + Demo	AI-assisted table merging	Organize tabular datasets
	17	Dates and Time Data	Coding	AI-generated time-	Work with date/time

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Types	Practice	series examples	data
	18	Categorical Data Handling	Lecture + Examples	AI-assisted category analysis	Represent discrete categories
7	19	Data Normalization Techniques	Lecture + Demo	AI-assisted preprocessing	Normalize datasets
	20	Missing Data Handling	Practical Session	AI-generated interpolation methods	Handle incomplete data
	21	Correlation and Polynomial Fitting	Problem Solving	AI-assisted curve fitting	Analyze relationships
8	22	User Interaction in MATLAB	Coding Demo	AI-generated interaction examples	Develop interactive programs
	23	Decision Branching	Lecture + Examples	AI-assisted flow control	Implement conditions
	24	For Loops and While Loops	Practical Coding	AI-generated loop tracing	Automate repetitive tasks
9	25	Creating Functions in MATLAB	Lecture + Demo	AI-assisted function generation	Build reusable functions
	26	Function Files and Workspaces	Coding Practice	AI-generated debugging tips	Manage function files
	27	Debugging MATLAB Code	Practical Session	AI-assisted debugging	Troubleshoot MATLAB programs
10	28	MATLAB Onramp Overview	Lecture + Demo	AI-guided tutorials	Understand MATLAB Onramp
	29	MATLAB Editor and Scripts	Coding Practice	AI-generated script optimization	Develop efficient scripts
	30	Vector and Matrix Calculations	Practical Session	AI-assisted matrix operations	Apply array calculations
11	31	Array Indexing and Modification	Lecture + Examples	AI-assisted indexing support	Manipulate arrays efficiently
	32	Plotting and Annotation	Coding Demo	AI-generated plot annotations	Create informative plots
	33	Importing Data into MATLAB	Practical Session	AI-assisted import tools	Import engineering datasets
12	34	Logical Arrays and	Lecture +	AI-generated logic	Use logical arrays

Department of Computer Science & Medical Engineering

Week	Lecture No.	Topics Covered	Teaching Methods	AI Integration	Expected Learning Outcome
		Programming Constructs	Demo	examples	
	35	Decision Making and Loops	Coding Practice	AI-assisted flow analysis	Implement program logic
	36	MATLAB Final Project Discussion	Project-Based Learning	AI-supported project ideas	Design MATLAB projects
13	37	Introduction to Image Processing	Lecture + Demo	AI-assisted image analysis	Understand image fundamentals
	38	Image Segmentation Techniques	Coding Demo	AI-generated segmentation methods	Segment images
	39	Preprocessing and Postprocessing	Practical Session	AI-assisted filtering	Enhance image quality
14	40	Object Detection and Edge Detection	Lecture + Coding	AI-assisted feature extraction	Detect image features
	41	Batch Image Processing	Coding Practice	AI-generated automation scripts	Process multiple images
	42	Image Registration Techniques	Lecture + Demo	AI-assisted registration	Align images accurately
15	43	Case Study Discussion	Discussion Session	AI-generated case studies	Analyze real-world applications
	44	Course Revision	Interactive Session	AI-generated quizzes	Consolidate MATLAB concepts
	45	Final Problem Solving Session	Q&A + Coding	AI-assisted practice problems	Apply complete course knowledge

****

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING
Devarakaggalahalli, Harohalli, Kanakapura Road, Bengaluru South, Karnataka – 562 112
Department of Computer Science & Medical Engineering

R PROGRAMMING FOR DATA ANALYTICS

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

SEMESTER – VI

Subject Code	:	xxxxxx	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39
L–T–P	:	3–0–0-0			

Course Learning Objectives:

This Course will enable students to:

1. **Understand the fundamentals of R programming** including syntax, data types, operators, control structures, functions, and packages to build a strong foundation for data analysis.
2. **Explore, manipulate, and visualize datasets** using R libraries like `ggplot2` and `dplyr` to derive meaningful insights from raw data.
3. **Perform Exploratory Data Analysis (EDA)** to identify trends, patterns, anomalies, and relationships in datasets, while understanding key assumptions for statistical modeling.
4. **Apply regression techniques** such as linear, multiple, and non-linear regression to analyze relationships between variables and evaluate model performance.
5. **Implement classification algorithms** including logistic regression, SVM, KNN, Naïve Bayes, Decision Trees, and Random Forests to solve real-world classification problems and interpret results.
6. **Evaluate models** using appropriate performance metrics and validation techniques to ensure robustness and generalization of analytical solutions.
7. **Develop critical thinking and problem-solving skills** by applying data science workflows to practical case studies using R.

Teaching-Learning Process (General Instructions)

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

9. ***Lecture method*** includes traditional and modern teaching techniques to explain deep learning concepts clearly. Different approaches like ***presentations and coding demos*** are used for better understanding.
10. ***Interactive teaching*** methods like group discussions, brainstorming, and peer learning are encouraged. These activities help students stay engaged and actively participate in the learning process.
11. Show ***Video/animation***to explain complex topics like 3D visualization in medical imaging and simulations.
12. Encourage ***Collaborative*** (Group Learning) Learning in the class.
13. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
14. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
15. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	08Hours
INTRODUCTION TO R Programming R Studio, R vs Python, Comments in R, Constants, R Datatypes, R Operators, R Conditional constructs, Looping constructs, Unconditional constructs, R Functions, Data structures used in R, Packages in R. **Engineering Applications:** Statistical computing in engineering research, automation of data analysis tasks, scientific simulations, signal and sensor data processing, bioinformatics analysis, financial engineering computations, and rapid prototyping of machine learning algorithms using R programming.	
UNIT – II	**08Hours**
Introduction to Data Analysis **08 Hours** erview of data analysis, working with directory in R, Loading and handling data in R, Data Visualization with ggplot2, Data Transformation with dplyr. **Engineering Applications:** lustrial process monitoring, business intelligence dashboards, healthcare data analytics, IoT sensor data analysis, quality control systems, energy consumption analysis, predictive maintenance systems, and visualization of engineering datasets using ggplot2 and dplyr.	
UNIT – III	**08 Hours**
Exploratory Data Analysis Exploring a new dataset, Anomalies in numerical data, visualizing relations between variables, Assumptions of Linear Regression, Validating Linear Assumption, Missing Values, Covariation, Patterns and Models, gglot2 Calls **Engineering Applications:** Detection of anomalies in manufacturing systems, medical diagnosis data exploration, environmental data analysis, fault detection in communication systems, traffic and transportation analytics, sensor pattern recognition, and preprocessing of engineering datasets for machine learning applications.	
UNIT – IV	**07Hours**
Regression Analysis Introduction, Types of Regression Analysis Models, Linear Regression, Simple Linear Regression, Non-Linear Regression, Regression Analysis with Multiple Variables, Cross Validation, Principal Component Analysis, Factor Analysis. **Engineering Applications:** Sales and demand forecasting, predictive maintenance in industries, healthcare risk prediction, energy load forecasting, weather prediction systems, engineering design optimization, reliability analysis, and performance prediction in mechanical and electrical systems using regression models.	
UNIT – V	**08 Hours**
Classification Introduction, Different types of Classification, Logistic Regression, Support Vector Machines, K-Nearest Neighbours, Naïve Bayes Classifier, Decision Tree Classification, Random Forest Classification, and Evaluation. **Engineering Applications:** Disease classification in healthcare, spam email filtering, image and speech recognition, fraud detection systems, industrial fault classification, customer segmentation, cybersecurity threat detection, predictive analytics in smart cities, and AI-driven decision support systems using machine learning classification algorithms.	

CO	Course Outcomes	Bloom taxonomy level
CO1	Describe the basic syntax, data structures, and programming constructs in R.	L2
CO2	Apply R for data analysis, transformation, and visualization using libraries like dplyr and ggplot2.	L3
CO3	Perform exploratory data analysis and handle data anomalies, missing values, and covariation in datasets.	L4
CO4	Develop regression models using R and apply techniques like PCA and factor analysis for dimensionality reduction.	L5
CO5	Build and assess classification models using various algorithms including SVM, KNN, Naïve Bayes, and Random Forest.	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	3
CO1	3	2	2	1	2				1	1			3	2
CO2	3	3	3	2	2				1	1			3	3
CO3	3	3	3	2	3				2	1			3	3
CO4	3	3	3	2	3				2	2			3	3
CO5	3	3	3	3	3				2	2			3	3

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

Activity Based Learning (Suggested Activities in Class)

1. Analyze research problems by reading research articles, group discussion, and presentations.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects.

Text books:

1. Hadley Wickham and Garrett Grolemund, “R for Data Science”, O’reilly, 2017.

REFERENCE BOOKS:

1. Dr. Bharati Motwani, “Data Analytics using R”, Wiley, 2019

Web links and Video Lectures (e-Resources):

1. https://r4ds.had.co.nz/
2. https://swirlstats.com/
3. https://www.tidyverse.org/
4. https://online-learning.harvard.edu/course/exploratory-data-analysis
5. https://www.kaggle.com/code?search=R

Activity Based Learning (Suggested Activities in Class)

1. R Programming Hackathon (Live Coding Challenge)
2. Data Visualization Project with ggplot2
3. Exploratory Data Analysis (EDA) Mini Project
4. Classification Model Building & Evaluation

****