DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Department of Computer Science and Engineering (Cyber Security)

SCHEME FOR BACHELOR OF TECHNOLOGY (B. Tech)

Batch 2025-2029

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

SCHEME - B. TECH – III SEM

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Examination (Maximum Marks)		
			L	T	P	J	C	CIE	SEE	TM
1	25CY2301	Applied Statistics & Stochastic Processes	3	0	0	0	3	60	40	100
2	25CY2302	Data Structures and Algorithms	3	0	2	0	4	60	40	100
3	25CY2303	Digital Logic and Computer Organization	3	0	2	0	4	60	40	100
4	25CY2304	Computer Networks	3	0	2	0	4	60	40	100
5	25CY2305	Cyber Security Essentials *	3	0	0	0	3	60	40	100
6	25CY2306	Sustainability Engineering & Environmental Social and Governance	2	0	0	0	2	60	40	100
7	25CY23XX	Skill Enhancement Course-I (Choice of Courses)	1	0	2	0	2	100	--	100
8	25CY2307	Cognitive and Technical Skills-III		-	-	-	P/F	-	-	--
			18	0	8	0	22	460	240	700

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,

SEE- Semester End Examinations, TM – Total Marks.

* The syllabus shall be furnished by Cyberbit

** The syllabus shall be furnished by NVIDIA

Skill Enhancement Course-I	
COURSE CODE	**COURSE NAME**
25CY2308	Discrete Mathematics
25CY2309	Linux Programming

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

SCHEME - B. TECH –IV SEM

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Examination (Maximum Marks)		
			L	T	P	J	C	CIE	SEE	TM
1	25CY2401	Differential Equations & Optimization	3	0	0	0	3	60	40	100
2	25CY2402	Cloud Security*	3	0	0	0	3	60	40	100
3	25CY2403	Embedded System Design	3	0	2	0	4	60	40	100
4	25CY2404	Design Studio II: Open Source Tools in Cyber Security	1	0	0	2	2	60	40	100
5	25CY2405	Introduction to Data Science	2	0	0	0	2	60	40	100
6	25CY2406	AI Essentials for Cyber Security*	3	0	2	0	4	60	40	100
7	25LSXXXX	Liberal Studies (Humanities / Social Science)	2	0	0	0	2	100	-	100
8	25CY24XX	Skill Enhancement Course II	1	0	2	0	2	100	-	100
9	25CY2407	Cognitive and Technical Skills-IV		-	-	-	P/F	-	-	--
			19	0	4	2	22	560	240	800

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,

SEE- Semester End Examinations, TM – Total Marks.

Liberal Studies	
Course Code	**Course Name**
25LS0001	Introduction to Drama
25LS0002	Introduction to Dance
25LS0003	Introduction to Music
25LS0004	Introduction to Photography
25LS0005	Introduction to Japanese language
25LS0006	Law for Engineers
25LS0007	Introduction to Painting
25LS0008	Communication Through Sanskrit
25LS0009	Vedic Mathematics
25LS0010	Fundamentals of Critical Thinking
25LS0011	Introduction to Film Studies
25LS0012	Practicing Yoga & Meditation
25LS0013	Cyber Crimes, Policies & Laws
25LS0014	Holistic Medicine
25LS0015	3-D Modelling using Tinkercad
25LS0016	Introduction to German Language
25LS0017	Design Thinking
25LS0018	Environmental Ethics & Climate Responsibility
25LS0019	Psychology for Engineers
25LS0020	Indian Knowledge Systems for Engineers
25LS0021	Food & Nutrition
25LS0022	Culture of Cities

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

Skill Enhancement Course II	
COURSE CODE	**COURSE NAME**
25CY2408	Web Technologies
25CY2409	DevOps

APPLIED STATISTICS & STOCHASTIC PROCESSES

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

SEMESTER - III

Course Code: 25CY2301	**Credits:** 03
Hours / Week: 03	**Total Hours:** 45
L–T–P–J :3–0–0–0	**CIE+SEE:**60+40 Marks

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. **Discuss** the fundamentals of stochastic processes and their applications in modeling random systems, decision-making, and AI-driven applications.

Teaching-Learning Process (General Instructions)

Following are sample new pedagogical methods, where teachers 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.

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

Authentic Intelligence:

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

AI Integration:

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

Course Outcome

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

Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Gupta, S. C., and V. K. Kapoor. Fundamentals of Applied Statistics. S. Chand, 2006.
2. Miller, Irwin, Marylees Miller, and John E. Freund. Miller and Freund'Probability and Statistics for Engineers. 8th ed., Pearson, 2018.
3. Mendenhall, William, Robert J. Beaver, and Barbara M. Beaver. Introductioto Probability and Statistics. Cengage Learning, 2020.
4. Ross, Sheldon M. Stochastic Processes. 2nd ed., John Wiley & Sons, 1995.
5. Operations Research: An Introduction Taha, Hamdy A. Operations Research: An Introduction. 10th ed., Pearson Education, 2017.

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. Mark L. Berenson, David M. Levine, Kathryn A. Szabat, Basic Business Statistics Concept and Application, Thirteenth Edition, Pearson Education Limited, 2015.

6. Bruce L. Bowerman, Richard T. O'Connell, Emily S. Murphree, Business Statistics in Practice, Seventh Edition, McGraw-Hill, 2014.
7. Douglas C. Montgomery, George C. Runger, Applied Statistics and Probability for Engineers, Sixth Edition, John Wiley and Sons, 2014.
8. Norean R. Sharpe, Richard D. De Veaux, Paul F. Velleman, Business Statistics, Fourth Edition, Pearson Education. 2019.

E-RESOURCES:

RECOMMENDED TOOLS

- Python
- R Programming

LAB EXERCISES

S. No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Random Variables and Probability Distributions (03 Hours)					
1	Random Variable Visualization	Generate and visualize discrete and continuous random variables	R Programming	1 h	**CO1**
2	PMF, PDF & CDF	Compute PMF, PDF, and CDF for standard distributions	R Programming	1 h	**CO1**
3	Joint Distributions	Simulate joint probability distributions	R Programming	1 h	**CO1**
MODULE II: Exercises 4–6: Bi-variate Data, Correlation and Regression Analysis (03 Hours)					
4	Bivariate Data Visualization	Visualize bivariate data using scatter plots	R Programming	1 h	**CO2**
5	Correlation Analysis	Compute Karl Pearson's and Spearman's rank correlation	R Programming	1 h	**CO2**
6	Linear Regression	Fit and evaluate regression model	R Programming	1 h	**CO2**
MODULE III: Exercises 7–9: Inference for Decision Making – I (03 Hours)					

7	Z-Test & t-Test	Perform Z-test and t-test (single & two-sample)	R Programming	1 h	**CO3**
8	Proportion Testing	Test for proportion and difference of proportions	R Programming	1 h	**CO3**
9	F-Test & ANOVA	Perform F-test and one-way ANOVA	R Programming	1 h	**CO3**
MODULE IV: Exercises 10–12: Inference for Decision Making – II (03 Hours)					
10	Chi-Square Tests	Perform goodness of fit and independence tests	R Programming	1 h	**CO4**
11	Bayesian Inference	Implement basic Bayesian inference techniques	R Programming	1 h	**CO4**
12	Data Visualization	Visualize probability distributions and statistical results	R Programming	1 h	**CO4**
MODULE V: Exercises 13–15: Stochastic Processes (03 Hours)					
13	Monte Carlo Simulation	Perform Monte Carlo simulations for probabilistic models	R Programming	1 h	**CO5**
14	Markov Chains	Construct transition matrices and simulate state transitions	R Programming	1 h	**CO5**
15	Exponential Distribution	Study exponential distribution and waiting time analysis	R Programming	1 h	**CO5**

DATA STRUCTURES AND ALGORITHMS

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

SEMESTER – III

Course Code: 25CY2302	Semester:3
Prerequisites: Introduction to C Programming	Hours / Week:4 Hours
Course Type: Core	Credits:04
L – T – P – J: 3 – 0 – 2 – 0	Total Hours:45+30 Hours

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

MODULE – 1: Foundations of Algorithms	08 Hours
Algorithm Basics, Performance Analysis & Recurrence Relations **Topics:** 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.	
AI Integration & Authentic Intelligence Reflection (AI² Pedagogy): **AI Integration:** Use USF Data Structure Visualizations to explore algorithm performance visually: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html **Authentic Intelligence Reflection:** Reflect on how asymptotic notation helps in making design decisions; critically evaluate the efficiency of a chosen algorithm using Big-O analysis.	
MODULE – 2: Basic Data Structures	10 Hours
Arrays, Pointers, Linked Lists & Divide and Conquer **Topics:** 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.	
AI Integration & Authentic Intelligence Reflection (AI² Pedagogy): **AI Integration:** USF Data Structure Visualizations — linked list module: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html **Authentic Intelligence Reflection:** In module 2 the visualization tools are used to verify understanding of the core concept that includes linked list. The visualization tools act as a powerful feedback mechanism to enhance learning. By leveraging such tools, learners can experience adaptive feedback for continuous improvement.	
MODULE – 3: Stacks, Queues, and Trees	08 Hours
Stacks, Queues, Expression Evaluation & Tree Structures **Topics:** 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.	
AI Integration & Authentic Intelligence Reflection (AI² Pedagogy): **AI Integration:** USF Data Structure Visualizations — stack, queue, and tree modules: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html **Authentic Intelligence Reflection:** In module 3 the visualization tools are used to verify understanding of the core concepts that includes stack, queue and trees. The visualization tools act as a powerful feedback mechanism to enhance learning. By leveraging such tools, learners can experience adaptive feedback for continuous improvement.	
MODULE – 4: Algorithm Design Techniques	09 Hours

Brute Force, Graph Traversal & Hashing **Topics:** 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.	
AI Integration & Authentic Intelligence Reflection (AI² Pedagogy): **AI Integration:** USF Data Structure Visualizations — graph and hashing modules: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html **Authentic Intelligence Reflection:** In module 4 the visualization tools are used to verify understanding of the core concepts that includes graphs. The visualization tools act as a powerful feedback mechanism to enhance learning. By leveraging such tools, learners can experience adaptive feedback for continuous improvement.	
MODULE – 5: Advanced Algorithms	10 Hours
Greedy Methods, Dynamic Programming & Backtracking **Topics:** 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.	
AI Integration & Authentic Intelligence Reflection (AI² Pedagogy): **AI Integration:** USF Data Structure Visualizations — algorithm design modules: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html **Authentic Intelligence Reflection:** Reflect on the trade-offs between greedy and dynamic programming approaches for the knapsack problem; critically evaluate when backtracking is a practical choice vs. other design strategies.	

Course Outcome:

CO	Course Outcome	Blooms Level
CO1	Analyze the efficiency of algorithms using asymptotic notations, recurrence relations, and mathematical techniques to evaluate computational complexity.	L4
CO2	Implement linear data structures such as arrays, linked lists, stacks, and queues to solve computational problems efficiently.	**L3**
CO3	Apply tree-based data structures and traversal algorithms to organize, search, and process hierarchical data.	**L3**
CO4	Identify solutions using fundamental algorithm design techniques including divide-and-conquer, graph traversal, hashing, and brute-force approaches.	**L3**
CO5	**I**mplement advanced algorithmic strategies such as greedy methods, dynamic programming, and backtracking to solve optimization and decision problems while distinguishing complexity classes (P, NP, NP-Hard, NP-Complete).	**L3**

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

Text Books

1. **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein**, *Introduction to Algorithms*, 4th Edition, MIT Press, 2022.
2. **Ellis Horowitz, Sartaj Sahni, and Dinesh Mehta**, *Fundamentals of Data Structures in C++*, 2nd Edition, Universities Press.

Reference Books

1. **Mark Allen Weiss**, *Data Structures and Algorithm Analysis in C++*, 4th Edition, Pearson Education.
2. **Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman**, *Data Structures and Algorithms*, Addison-Wesley.
3. **Robert Sedgewick and Kevin Wayne**, *Algorithms*, 4th Edition, Addison-Wesley.

4. **Steven S. Skiena**, *The Algorithm Design Manual*, 3rd Edition, Springer.
5. **S. Dasgupta, C. Papadimitriou, U. Vazirani**, *Algorithms*, McGraw-Hill Education.

DATA STRUCTURES LABORATORY

Total Contact Hours: 30 Hours

Following are experiments to be carried out using either C programming language.

1. To Implement C programs to perform array operations.
2. To determine the validity of a 9x9 Sudoku board (application of 2-dimensional array).
3. To store, retrieve and update the elements in structures (structures and pointers to structures).
4. To implement stack using linked list.
5. To implement a queue data structure using a singly linked list.
6. To implement a singly linked list and its operations.
7. To implement a doubly linked list and its operations.
8. To create a circular queue using a circular linked list data structure
9. To implement binary tree traversal techniques.

OPEN-ENDED EXPERIMENTS

1. Design a web browser history tracker in C. Implement a stack data structure to keep track of visited URLs. Create functions to push new URLs onto the stack as users visit websites and pop URLs when users navigate backward in their browsing history.
2. Imagine you are responsible for designing a queue-based system to manage the queue of regular customers waiting to purchase cinema tickets at a popular movie theatre. Your system should ensure fair and efficient ticket sales for all customers. When a customer arrives at the cinema, they join the queue. Each customer is represented by his name, age (for record-keeping), and number of tickets needed. When a customer reaches the front of the queue, they are served by the ticketing agent. Implement a ticket sale process where the agent provides the customer with the requested ticket(s). Initialize the total number of tickets and if the tickets are sold, then the ticketing agent should display a houseful message.

Digital Logic and Computer Organization

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

SEMESTER -III

Course Code: 25CY2303	**Credits:** 04
Hours / Week: 03	**Total Hours:**45+30
L–T–P–J:3–0–2–0	**CIE+SEE :**60+40 Marks

Course Vision:

To provide a strong foundation in digital logic and computer organization, enabling students to understand, design, and analyze modern computing systems, while developing problem-solving skills and the ability to apply core concepts in real-world hardware and system-level applications.

Course Objectives:

This course will enable students:

1. To introduce the fundamentals of Verilog HDL, including syntax, operators, and coding structure.

2. **To provide** a foundation for advanced studies in VLSI design, embedded systems, processor architecture, and intelligent digital hardware systems.

3. **To understand** the fundamentals of sequential circuits including latches, gated latches, and various types of flip-flops such as SR, D, JK, T, and master-slave JK flip-flops.

4. **To understand** the principles and organization of cache memory and its role in improving processor performance.

5. **To study** different instruction formats and addressing modes used in computer architecture for efficient data access and execution.

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**08 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. Introduction to GPU Programming model, CPU vs GPU algorithm, GPU architecture, Basics of CUDA program.

Applications:

- Used in computers and digital electronics for data representation.
- Hexadecimal numbers are used in memory addressing and programming.
- Binary conversions are essential in communication and embedded systems.

AI Integration:

- AI automatically generates test benches.
- Machine learning detects hardware bugs quickly.
- AI improves testing accuracy and speed.
- Used in smart semiconductor testing platforms.

Authentic Intelligence:

- AI optimizes digital circuit design and Boolean simplification.
- Used in intelligent chip design and FPGA development.
- AI-assisted tools automatically generate Verilog code.
- Helps in automatic testing and hardware verification.

MODULE – II	**09 Hours**

Combinational Circuit Building

Topics: 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

Applications:

- Used in Arithmetic Logic Units (ALU) for arithmetic and logical operations.
- Used in data routing through multiplexers and demultiplexers.
- Used in memory address decoding in computer systems.
- Used in communication systems for encoding and decoding data.

AI Integration:

- Leverage the NVIDIA CUDA Compiler (NVCC) in conjunction with AI-driven optimization like ChatGPT 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.

Authentic Intelligence:

- 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

MODULE – III	**12 Hours**

Sequential Circuits-1

Topics:

Sequential Circuits-1

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. Binary counters–asynchronous and synchronous, mod-n counter, Verilog blocking and non-blocking. Introduction to Electronic Design Automation: FPGA Design Flow, ASIC Design flow.

Applications:

- Used in memory storage elements and processor registers.
- Shift registers are used in serial and parallel data communication.
- Flip-flops are widely used in counters, timers, and control systems.
- Setup time, hold time, and propagation delay are important in high-speed processor and embedded system design.

AI Integration:

- 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:

- 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.

MODULE – IV	**08 Hours**

Introduction to Computer Architecture

Topics: Defining Computer architecture, Trends in Technology, Power, and energy in Integrated Circuits. Measuring, Reporting, and Summarizing Performance. Quantitative Principles of Computer Design. Classifying Instruction Set Architectures. Memory Addressing. Type and Size of Operands. Operations in the Instruction. Instructions for Control Flow.

Applications:

- Increases processor performance by reducing memory access time
- Used in modern CPUs and GPUs for fast computation
- Supports multitasking and high-speed applications
- Improves execution speed in operating systems and databases

AI Integration:

- Cache memory accelerates AI model execution and inference.
- DRAM supports high-speed training of machine learning models.

- SSDs improve AI dataset loading and processing performance.
- RAID systems provide reliable storage for large-scale AI data.

Authentic Intelligence:

- Intelligent systems require fast and reliable memory access mechanisms.
- Cache and DRAM improve real-time intelligent decision-making.
- RAID and SSD ensure secure and trustworthy intelligent data storage.

MODULE – V	**08 Hours**

Pipelining and Memory Hierarchy:

Topics: The Major Hurdle of Pipelining—Pipeline Hazards. How Is Pipelining Implemented? What Makes Pipelining Hard to Implement? Introduction to memory hierarchy. Cache Performance. Six Basic Cache Optimizations. Virtual Memory. Protection and Examples of Virtual Memory.

Applications:

- Controls processor operations and execution flow
- Supports communication between hardware and software
- Used in operating systems and embedded systems
- Essential for assembly language programming

AI Integration:

- Instruction pipelining improves AI computation speed and efficiency.
- DMA enables fast transfer of large AI datasets.
- RISC processors are widely used in AI-enabled embedded systems.
- Interrupt-driven I/O supports real-time AI and robotic applications.

Authentic Intelligence:

- Intelligent systems require efficient instruction execution and data transfer.
- Pipelining supports real-time intelligent decision-making systems.
- DMA and interrupts improve responsive intelligent computing environments.
- RISC and CISC architectures support reliable AI and embedded applications.

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Interpretation of Boolean Expressions of digital design in simplified form.	L2
2	Build the various elements of digital logic system with Verilog.	L3
3	Design Combinational and Sequential logic circuits and analyse the hardware model of a digital system at different levels of abstraction in Verilog	L5
4	Understand the basic principles of the quantitative design of computers and Instruction set architecture.	L2

5	Analyse the implementation of pipelined processors and cache optimization	L3

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

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.

4. John L. Hennessy and David A. Patterson. Computer Architecture A Quantitative Approach. Sixth edition, Morgan Kauffmann publisher .

REFERENCE BOOKS:

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.

E-Resources:

https://archive.nptel.ac.in/courses/106/105/106105165/

https://nptel.ac.in/courses/117105080

Recommended Tools

- Digital Logic Design & Simulation
- HDL Design & Simulation Tools
- FPGA & Hardware Implementation
- Processor & Architecture Simulation

Laboratory Component - 30 Hours

Part A: Implementation using 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 using CUDA 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 using CUDA 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.

COMPUTER NETWORKS

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

SEMESTER - III

Course Code: 25CY2304	**Credits:**04
Hours / Week: 04	**Total Hours:**45+30
L–T–P–J: 3–0–2–0	**CIE+SEE:** 60+40 Marks

Course Vision:

To empower students with the knowledge and skills required to understand, design, analyze, and implement computer networks, enabling them to address real-world communication challenges using modern tools and technologies.

Course Objectives:

This course will enable students:

1. Understand layered network architecture and communication models.
2. Analyze data transmission and error control mechanisms
3. Design routing and IP addressing schemes
4. Evaluate transport protocols and congestion control
5. Explore application layer protocols and real-world networking

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**09 Hours**
Introduction: Data Communications, Networks, Network Types, Network Models, Protocol Layering, Reference Models: The OSI Reference Model, The TCP/IP Reference Model, Physical Layer: Data and signals Digital Transmission, (D-D Conversion) Bandwidth Utilization, Multiplexing, Switching, Circuit Switched Networks, Packet Switching. **Applications:** Internet backbone, Telecom systems **AI Integration: Wireshark, Cisco Packet** Tracer **Authentic Intelligence:** Analyze latency differences in switching	
MODULE – II	**09 Hours**
Data Link Layer: Link Layer Addressing, Error Detection and Correction, Block Coding, Cyclic Codes, Checksum. Data Link Control: DLC Services, Data-Link Layer Protocols, Media Access Control, Wired LANs, Ethernet protocol. **Applications:** LAN networks, Ethernet communication. **AI Integration**: NS2/NS3, MATLAB **Authentic Intelligence:** Study packet loss and retransmissions	
MODULE – III	**09 Hours**
Network Layer: Network Layer Services, Packet Switching, Network Layer Performance, IPV4 Addresses. Network Layer Protocols: Internet Protocol, ICMPV4, Unicast Routing, Routing algorithms, Unicast routing protocols, Internet Structure, Routing Information Protocol (RIP), Next Generation IP: IPV6 Addressing, IPV6 Protocol, ICMPv6 Protocol, Transition from IPV4 to IPV6, Congestion Control Algorithms, QoS. **Applications:** Internet routing, Cloud networking **AI Integration:** Cisco Packet Tracer, Python **Authentic Intelligence**: Case study on congestion	
MODULE – IV	**09 Hours**
Introduction and Transport-Layer Services: Overview of the Transport Layer in the Internet,: Connectionless Transport: UDP, UDP Segment Structure, UDP Checksum, Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols: Go-Back-N, Selective repeat, Connection-Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control: The Causes and the Costs of Congestion, Approaches to Congestion Control. **Applications:** Streaming services, Online applications **AI Integration:** Wireshark, Socket programming **Authentic Intelligence**: Compare TCP vs UDP performance	

MODULE – V	**09 Hours**
Application Layer: Introduction, client server programming, WWW (World Wide Web) and HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), E-mail, telnet, DNS (Domain Naming System) SNMP (Simple Network Management Protocol). **Applications:** Web systems, Email services **AI Integration:** Postman, MQTT tools **Authentic Intelligence:** Build client-server application	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain network models and physical layer transmission techniques.	L2, L3
2	Analyze error control techniques and data link layer protocols.	L3, L4
3	Design IP addressing and evaluate routing algorithms.	L3, L4, L5
4	Evaluate transport layer protocols and congestion mechanisms.	L3, L4, L5
5	Analyze application layer protocols and services.	L3, L4

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Behrouz A. Forouzan—Data Communications and Networking‖, TataMcGraw-Hill,5th Edition, 2012.
2. Andrew S. Tanenbaum, David.J. Wetherall, —Computer Networks‖, Prentice-Hall, 5thEdition 2010.
3. James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth edition, Pearson, 2017.

REFERENCE BOOKS:

1. Data and Computer Communication William Stallings 8th Edition Pearson Education 2008.
2. Computer Networks – A Systems Approach Larry L. Peterson and Bruce S. Davie 4th Edition Elsevier 2007

E-BOOKS:

1. An Introduction to Computer Networks Peter L Dordal 1st Edition - 2020
   https://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

2. A Top-Down Approach: Computer Networking James F Kurose & Keith W Ross

   8th Edition Pearson

   https://gaia.cs.umass.edu/kurose_ross/online_lectures.htm

LAB EXERCISES -30 Hours

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
1	Ring Topology and Bus Topology	Implement Ring topology and Bus topology operation using NS2.	NS2/NS3	2 h	CO1
2	Three node point to point network	Implement three nodes point – to – point networks with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.	NS2/NS3	2 h	CO1
3	Ethernet LAN using n-nodes with multiple traffic	Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source /destination.	NS2/NS3	2 h	CO1
4	Simple ESS with wireless LAN	Implement simple ESS and transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.	NS2/NS3	2 h	CO2
5	Operation of Stop and wait Protocol	Write a NS2 script to implement the operation of Stop and Wait protocol	NS2/NS3	2 h	CO4

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
6	Error detection using CRC-CCITT	Write a program for error detecting code using CRC-CCITT (16-bits).	C, C++, Java, Python	2 h	CO2
7	Determination of shortest path using Bellman-Ford algorithm	Write a program to find the shortest path between vertices using Bellman-Ford algorithm.	C, C++, Java, Python	2 h	CO3
8	Congestion control using Leaky bucket concept	Write a program for congestion control using leaky bucket algorithm.	C, C++, Java, Python	2 h	CO3
9	Client/ Server Communication using TCP/ IP Socket	Write a program to Implement TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present	C, C++, Java, Python	2 h	CO4
10	DNS Server	Implement a simple DNS server to resolve the IP address for the given domain name	C, C++, Java, Python	2 h	CO5

SCHOOL OF ENGINEERING

(A State Private University under the Karnataka Act No. 20 of 2013)

Approved by UGC & AICTE, New Delhi, Accredited by NAAC with A+ grade

CYBER SECURITY ESSENTIALS

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

SEMESTER - III

Course Code: 25CY2305	**Credits:**03
Hours / Week: 03	**Total Hours:**45
L–T–P–J: 3–0–0–0	**CIE+SEE:**60+40 Marks

Course Vision:

To develop foundational and applied competencies in cybersecurity, enabling learners to understand, analyze, and mitigate cyber threats using modern tools, ethical practices, and intelligent technologies.

Course Objectives:

The course aims to enable students to:

1. To introduce core principles of cybersecurity and threat landscape
2. To develop skills in securing systems, networks, and applications
3. To provide exposure to cyber laws, risk management, and ethical practices
4. To integrate AI-driven security approaches and authentic intelligence analysis
5. To prepare students for real-world cybersecurity problem-solving

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**09 Hours**

FUNDAMENTALS OF CYBER SECURITY

Topics: Fundamentals of Cyber Security: CIA Triad (Confidentiality, Integrity, Availability), Cyber Security Terminology and Concepts, Types of Cyber Threats: Malware (Virus, Worm, Trojan, Ransomware, Spyware), Phishing and Social Engineering Attacks, Insider Threats and Advanced Persistent Threats (APT), Attack Vectors and Surfaces, Cyber Kill Chain Model.

Applications: Enterprise Security Awareness Programs, Threat Modelling in Organizations, Incident Response Planning

AI Integration: AI-based phishing detection tools, Malware classification using ML models, Threat intelligence platforms (AI-assisted SIEM)

Authentic Intelligence: Human intuition in identifying social engineering, Analyst-driven threat interpretation beyond automated alerts, Ethical decision-making in cyber defense

MODULE – II	**09 Hours**

NETWORK SECURITY

Topics: Network Security Architecture and Security Concepts, OSI & TCP/IP Security Issues , Common Network Attacks: DoS/DDoS, Man-in-the-Middle (MITM) , Packet Sniffing & Spoofing, Firewalls: Types (Packet Filtering, Stateful, Proxy), Intrusion Detection and Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Wireless Network Security (WEP, WPA, WPA3).

Applications: Secure Network Design, Enterprise Firewall Configuration, Network Traffic Monitoring

AI Integration: AI-based intrusion detection systems, Network anomaly detection using ML, Automated log analysis tools

Authentic Intelligence: Human validation of anomaly detection results, Context-aware network defense strategies, Combining intuition with AI alerts

MODULE – III	**09 Hours**

CRYPTOGRAPHY & DATA SECURITY

Topics: Introduction to Cryptography, Symmetric Encryption (AES, DES), Asymmetric Encryption (RSA), Hash Functions (SHA family, MD5 limitations), Digital Signatures and Certificates, Basics of Public Key Infrastructure (PKI), Key Management and Distribution, Basics of Blockchain Security.

Applications: Secure Communication Systems, Digital Transactions and E-Commerce, Data Integrity Verification

AI Integration: AI-assisted cryptanalysis research tools, Blockchain analytics tools, Automated certificate management systems

Authentic Intelligence: Understanding limitations of encryption, Ethical considerations in cryptographic use, Human oversight in key management

MODULE – IV	**09 Hours**
SYSTEM SECURITY & RISK MANAGEMENT **Topics:** Operating System Security: Access Control Models (DAC, MAC, RBAC), Vulnerability Assessment and Penetration Testing (VAPT), Malware Analysis Basics, Risk Management: Risk Identification, Assessment, Mitigation, Security Policies and Governance, Backup and Disaster Recovery, Security Auditing. **Applications:** Enterprise Risk Management, Security Audits and Compliance, Incident Handling **AI Integration:** AI-based vulnerability scanners, Automated penetration testing tools, Risk prediction models **Authentic Intelligence:** Human judgment in risk prioritization, Ethical hacking decision-making, Interpretation of automated scan results	
MODULE – V	**09 Hours**
Cyber Laws, Ethics **Topics:** Cyber Laws: Information Technology Act, 2000, Data Protection and Privacy Laws, Cyber Ethics and Professional Responsibility, Digital Forensics Basics, Emerging Trends: Cloud Security, IoT Security, AI in Cybersecurity, Zero Trust Architecture, Cyber Warfare and National Security **Applications:** Legal Compliance in Organizations, Digital Forensic Investigations, Secure Cloud Deployment. **AI Integration:** AI-based forensic tools, Cloud security posture management tools, Threat intelligence platforms **Authentic Intelligence:** Ethical reasoning in cyber investigations, Human-led forensic interpretation, Balancing automation with accountability	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Understand** the core cybersecurity concepts, security principles (CIA triad), types of cyber threats, and basic cryptographic techniques for protecting information systems.	L3
2	**Apply** network security mechanisms such as firewalls, IDS/IPS, VPNs, and secure protocols to protect data communication and defend against network-based attacks.	L3
3	**Identify** system and application vulnerabilities, including operating system weaknesses and web application threats (e.g., OWASP Top 10), and recommend appropriate mitigation strategies.	L3

4	**Interpret** cyber laws, ethical practices, and risk management frameworks to ensure legal compliance and develop incident response strategies.	L3
5	**Analyse** emerging cybersecurity technologies such as cloud security, IoT security, AI in cybersecurity, and Zero Trust models to address modern threat landscapes.	L4

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Cybersecurity Essentials, Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, 1st Edition, Cisco Press (Pearson Education), 2018, ISBN: 978-1587134388
2. Introduction to Cyber Security, The Open University, Open Learn / The Open University.

REFERENCE BOOKS:

1. William Stallings, Lawrie Brown, Computer Security: Principles and Practice, 4th Edition, Pearson, 2018, ISBN: 978-0134794105.
2. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson, 2017, ISBN: 978-0134444284.
3. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook, 2nd Edition, Wiley, 2011, ISBN: 978-1118026472.
4. Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008, ISBN: 978-1593271442.

RECOMMENDED TOOLS

- Security Tools: Wireshark, Nmap, Metasploit, Burp Suite
- SIEM Tools: Splunk, IBM QRadar
- AI/ML Tools: Python (Scikit-learn, TensorFlow), Weka
- Cloud Security: AWS Security Hub, Azure Defender

Sustainability Engineering & Environmental Social and Governance

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

SEMESTER - III

Course Code:25CY2306	**Credits:**02
Hours / Week: 02	**Total Hours:**30
L–T–P–J:2–0–0–0	**CIE+SEE :**60+40 Marks

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 analyse 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. Analyse 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 Outcome

Course Outcomes (COs)	Description
1	Explain fundamental concepts of sustainability engineering, ESG framework, and analyse environmental impact factors such as energy consumption and resource usage in digital systems.
2	Apply sustainability principles, ESG indicators, and analytical models to evaluate environmental performance and efficiency in computing and engineering systems.

3	Evaluate energy usage patterns, system efficiency, and resource optimization in IT infrastructure such as cloud systems, data centers, and software applications.
4	Analyse sustainability metrics and lifecycle impacts of engineering systems using ESG reporting frameworks, carbon footprint estimation, and responsible design principles.
5	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.

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 analyse 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

MODULE – I
Sustainability Fundamentals & ESG **Topics:** Sustainable development, Triple Bottom Line, ESG framework, SDGs, ESG vs CSR, sustainability challenges in digital economy. **Engineering Applications:** Green computing, sustainable IT infrastructure, energy-efficient data centers. **AI Integration:** AI-based sustainability trend analysis, ESG scoring tools.
MODULE – II
Environmental Sustainability in Digital Systems **Topics:** Carbon footprint of IT systems, energy consumption in data centers, lifecycle assessment (LCA), e-waste management, green cloud computing. **Engineering Applications:** Energy-efficient servers, cloud optimization, sustainable software engineering. **AI Integration:** AI models for energy prediction and carbon footprint estimation.
MODULE – III
Social Sustainability & Ethical Computing **Topics:** Stakeholder analysis, digital inclusivity, data privacy, ethical AI, cybersecurity and society, social impact of automation. **Engineering Applications:** AI ethics in recommendation systems, fairness in algorithms, inclusive software design. **AI Integration:** Sentiment analysis, bias detection in AI systems.
MODULE – IV
Governance, ESG Reporting, Sustainable Design & Future Technologies **Topics:** Corporate governance, ESG reporting frameworks (GRI, TCFD), ESG metrics, sustainability KPIs, data visualization, Circular economy, sustainable software design, green AI, smart cities, Industry 4.0 sustainability, digital twins for sustainability **Engineering Applications:** Corporate ESG dashboards, compliance analytics, risk management systems, Smart grids, intelligent transportation, sustainable IoT systems. **AI Integration:** AI-based ESG report analysis, dashboards using Power BI/Python, AI-based optimization of sustainable systems.

MODULE – V

Energy Efficiency, Audit & Smart Energy Systems

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-based energy consumption prediction using machine learning models, Optimization of energy usage in data centers using AI algorithms.

AI Integration & Authentic Intelligence Reflection (AI² 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 4.0 systems, 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.

- **Authentic Intelligence Reflection:**
  - **Module 1**: Students critically evaluate AI-generated ESG scores, sustainability indicators, and SDG mappings, identifying inconsistencies, data gaps, and assumptions in AI-based assessments.
  - **Module 2:** Students verify AI-based predictions of carbon footprint and energy consumption in IT systems, assessing limitations in datasets, boundary conditions, and real-world applicability.
  - **Module 3:** Students analyse AI outputs for bias detection, fairness, and ethical implications, identifying potential risks in algorithmic decision-making and digital inclusivity.

- **Module 4:** Students validate AI-generated ESG reports, dashboards, and KPIs, critically examining data reliability, reporting assumptions, and compliance with standards (GRI, TCFD).
- **Module 5:** Students evaluate AI-based energy predictions, audit results, and optimization strategies, identifying inaccuracies in models, assumptions in energy savings, and practical feasibility.

Research Integration Teaching

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

- Green Data Centers and Sustainable Cloud Computing (energy-efficient computing, cooling systems, carbon reduction strategies)
- Carbon Footprint of AI and Digital Systems (energy consumption of AI models, emissions from data centers)
- Life Cycle Assessment (LCA) in IT Systems (environmental impact from hardware manufacturing to disposal)
- Sustainable Software Engineering and Green AI (energy-aware algorithms, low-power computing)
- ESG and Digital Transformation (role of ESG in technology companies, sustainability reporting and governance)

Students will review at least one recent research paper (journal/conference) in mentioned areas.

NOTE (AI² Pedagogy - Innovative Pedagogy):

1. Each module consists of AI usage, AI evaluation, and human judgment.
    - Students will use AI tools to analyze sustainability problems, such as carbon footprint estimation, ESG scoring, energy consumption prediction, and ethical AI assessment.
    - AI-generated outputs will be critically evaluated using sustainability principles, engineering logic, and real-world benchmarks (e.g., SDGs, ESG metrics, LCA results).
    - Strong emphasis will be placed on validating AI outputs through human judgment, including feasibility, ethical implications, and environmental impact.
    - Students will compare AI predictions with case studies, industry data, or standard frameworks (GRI, TCFD, SDGs) to ensure reliability.

0. Faculty must demonstrate AI use in class, show incorrect AI outputs, and guide students to critique.
    - Demonstration of AI tools for ESG scoring and sustainability analytics, Carbon footprint estimation of IT systems, Energy optimization in data centers and cloud systems, Ethical AI evaluation (bias detection, fairness analysis)
    - Faculty will intentionally present incorrect or biased AI-generated outputs, such as: Misleading ESG scores, Incorrect carbon emission estimates, Biased algorithmic decisions, Unrealistic energy optimization results.
    - Students will be guided to: Identify errors, biases, and incorrect assumptions Analyze limitations of AI models, propose corrected and improved solutions using engineering reasoning and sustainability frameworks.
0. Students must be evaluated on AI-assisted work, AI debugging, and decision-making Assignments will include:
    - AI-generated sustainability analyses (carbon footprint, ESG reports, energy audits) with mandatory validation
    - Case studies involving AI-based sustainability tools and dashboards
    - AI-assisted lifecycle assessment (LCA) and green computing analysis.

0. **Conclusion (Innovative Pedagogy Adopted)**:

. Teaching AI² Pedagogy:

. AI-assisted learning using analytical and sustainability tools

i. Students use AI tools for ESG analysis, carbon footprint estimation, energy prediction, and ethical AI assessment.

ii. Tools may include AI-driven dashboards, sustainability analytics platforms, and cloud-based simulation environments.

a. Problem-First Learning (Instead of Topic-First)

. Start with real-world sustainability challenges such as:

- "How can data centers reduce their energy consumption and carbon footprint?"
- "How can cloud computing systems be optimized for sustainability?"
- "How do AI systems impact fairness, privacy, and society?"
- "How can smart cities use IoT for efficient energy management?"

ii. Then introduce concepts and tools like ESG frameworks, carbon accounting, and sustainability metrics, Energy efficiency techniques in computing systems, Lifecycle assessment (LCA) and green computing principles, Ethical AI, bias detection, and governance frameworks.

c. Research-integrated teaching

- Research-oriented course delivery with exposure to modern sustainability applications. Topics include: Green computing and energy-efficient data centers, AI for sustainability and ESG analytics, Smart energy systems and sustainable IoT, Circular economy in digital systems
- Students work on challenges such as: Reducing carbon footprint of IT infrastructure Optimizing cloud resource allocation for energy efficiency, Improving ESG performance using data analytics

d. Sustainability Thinking

i. Energy Impact: Energy efficiency in computing systems, servers, and networks, Energy consumption in data centers and cloud platforms, Optimization of power usage using AI and smart systems

ii. Environmental Impact: Carbon emissions from IT infrastructure, E-waste management and sustainable hardware lifecycle, Environmental implications of large-scale AI models and data processing

iii. Lifecycle Cost and Sustainable Design: Cost-effective and energy-efficient system design, Lifecycle assessment of digital systems

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 OpenCourseWare – Courses on Sustainability, Energy, and Smart Systems
3. IEEE, Elsevier, and industry ESG reports for latest research and case studies

AI-Powered Assignments

AI² Pedagogy: *It is a structured teaching-learning approach where students use AI tools for problem-solving while simultaneously developing Authentic Intelligence through critical evaluation, ethical reasoning, and responsible decision-making.*

Assignment 1: AI-Based Energy System Efficiency Evaluation:

- Students use AI tools to analyze energy cycles in sustainability systems such as Data center cooling cycles, HVAC refrigeration systems, Renewable energy-based thermal cycles (e.g., Rankine cycle analogy in energy systems).
- Outputs are validated using energy balance principles, efficiency metrics and real-world benchmark data from sustainability reports. Focus is on energy efficiency and carbon reduction potential.

Assignment 2: AI Debugging in Engineering and Sustainability Systems

- Students are given AI-generated incorrect or biased solutions related to Energy consumption prediction models, ESG scoring or carbon footprint estimation, Flow or thermal system performance analysis.
- Tasks include Identifying logical, numerical, and assumption-based errors, detecting sustainability misinterpretations (e.g., ignoring emissions or losses), Correcting outputs using engineering reasoning and ESG frameworks

Assignment 3: Comparative AI Analysis for Sustainability Decision-Making

- Students compare outputs from multiple AI tools/models for a given sustainability engineering problem such as Energy optimization in cloud/data centers, Carbon footprint estimation of digital systems, Efficiency of cooling or flow-based infrastructure systems.
- Evaluation criteria include: Accuracy and consistency of AI outputs, Assumptions and model limitations, Reliability for real-world engineering decision-making, Environmental and ESG implications

Mini Project Component

- **Industry-Driven Projects**

The following projects simulate real-world engineering and sustainability challenges in digital, energy, and infrastructure systems, integrating AI-based analysis, validation, and decision-making.

Project 1 – Smart Data Center Energy Optimization System

- **Problem:** Design and analyse an energy-efficient data center system to minimize power consumption and carbon footprint.
- **Concepts Used:** Energy efficiency metrics (PUE, energy intensity), Carbon footprint estimation, Resource allocation and load balancing, green computing principles
- **Industry Relevance:** Cloud computing companies, IT infrastructure and data center management, Large-scale enterprise systems

Project 2 – Smart Energy Management System for Buildings/IoT

- **Problem:** Design and simulate a smart system to monitor and optimize energy usage in buildings using IoT and AI.
- **Concepts Used:** Energy audit and consumption analysis, IoT-based monitoring systems, Predictive analytics for energy optimization, Smart grid and automation concepts
- **Industry Relevance:** Smart cities and infrastructure, Building energy management systems, IoT and automation industries

Project 3 – Sustainable Cloud Computing Optimization

- **Problem:** Develop strategies to optimize cloud resource usage for reduced energy consumption and improved sustainability.
- **Concepts Used:** Virtualization and cloud resource allocation, Energy-efficient scheduling algorithms, green cloud computing principles, Lifecycle assessment of cloud services
- **Industry Relevance:** Cloud service providers (AWS, Azure, Google Cloud), IT service companies, SaaS and platform engineering

DISCRETE MATHEMATICS

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

SEMESTER - III

Course Code:25CY2308	**Credits: 02**
Hours / Week:02	**Total Hours:30**
L–T–P–J:1–0–2–0	

Course Vision:

To build a robust theoretical foundation in mathematical logic, counting, and structure that directly enables the analysis, design, and validation of secure computer systems.

Course Objectives:

This Course will enable students to:

1. Learn the set theoretic concept and its application in theory of computation.
2. Determine the concepts of mathematical induction, recursive relations and their application.
3. Illustrate the association of functions, relations, partial ordered set and lattices with problems related to theoretical computer science and network models.

Teaching-Learning Process (General Instructions)

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

1. **Lecture method:** along with traditional lecture method, different type of teaching methods may be adopted to attain the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**10 Hours**

SET THEORY: Sets and subsets, Operations on Sets: Basic set operations, algebraic properties of sets, The Addition Principle , **RELATIONS AND ITS PROPERTIES:** Relations and their properties, N-Ary Relations and their applications, Representing relations.

Textbook – 2: 1.1, 1.2; Textbook – 1: 7.1., 7.2, 7.3

Applications: Access Control Systems, Network Security (Firewall Rules), Malware Signature Detection, Vulnerability Management

AI Integration: Feature Engineering in ML, Clustering Attacks, Anomaly Detection

Authentic Intelligence: Efficient threat classification using minimal data, Logical reasoning for zero-trust architectures, Identifying overlaps between attack vectors

NVIDIA DLI Courses (Relevant to this Module):

1. **Accelerating End-to-End Data Science Workflows** (8 hrs) | Path: Data Science Training - Accelerated Data Science Foundations https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-01+V2

MODULE – II	**10 Hours**

RELATIONS AND ORDER RELATIONS: Closure of relations, Equivalence Relations, Partial Orderings, Functions, The Growth of Functions, Transitive Closure and Warshall's Algorithm.

Textbook – 1: 7.4., 7.5, 7.6, 3.2

Applications: Attack Path Discovery, Network Security Analysis, Trust Propagation

AI Integration: Graph traversal algorithms, Threat propagation modeling, Link prediction in security graphs.

Authentic Intelligence: Detecting multi-stage attacks, Understanding lateral movement in networks

MODULE – III	**10 Hours**

MATHEMATICAL INDUCTION AND RECURSION: Mathematical Induction, Recurrence Relations: Rabbits and the Fibonacci Numbers, The Tower of Hanoi, Code word Enumeration, Solving Linear Recurrence Relations, Basic Connectives and Truth Tables

Textbook-1: 4.1;6.1, 6.2;1.1

Applications: Algorithm correctness & analysis, Growth modeling, Forecasting

AI Integration: Formal verification, RNNs, sequences, NLP encoding, Rule-based AI

- **Authentic Intelligence:** Clear decision-making, Eliminating ambiguity in reasoning, Developing structured problem-solving strategies

NVIDIA DLI Free Courses (Relevant to this Module):

1. **Accelerating End-to-End Data Science Workflows** (8 hrs) | Path: Data Science Training - Accelerated Data Science Foundations

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-01+V2

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** the concepts of set theory, set operations, and relations to model, analyze, and solve practical problems in engineering and computational domains.	L3
2	**Make use of** properties of relations, partial orderings, functions, and growth of functions, along with algorithms like Warshall's Algorithm to analyze, and solve real-world and computational problems.	L3
3	**Utilize** mathematical induction, recursion techniques, linear recurrence relations, and logical connectives to model, analyze, and solve problems in engineering and computing contexts.	L3

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Kenneth H. Rosen, "Discrete Mathematics and its applications", Tata McGraw Hill, 2003.
2. Bernard Kolman, Robert C. Busby, Sharon Ross, "Discrete Mathematical Structures", 3rd Edition, PHI 2001.

REFERENCE BOOKS:

1. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", IV Edition, Pearson Education, Asia, 2002.
2. J. P. Tremblay, R. Manohar, "Discrete Mathematical Structures with applications to computer Science", Tata McGraw Hill, 1987.
3. J K Sharma, "Discrete Mathematics", 3rd edition, 2013, Macmillan India Ltd.

LINUX PROGRAMMING

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

SEMESTER – III

Subject Code	**: 25CY2309**	**Credits**	**: 02**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 30 Hours**
L–T–P–J	**: 1–0–2–0**		

Course Learning Objectives: This course will enable students to:

1. **Apply** core Linux concepts and basic system commands to navigate, manage files, and compare operating system environments (Linux vs Unix vs Windows).
2. **Implement** Linux installation and configuration on virtual machines and **operate** system utilities for managing user sessions and desktop environments.
3. **Execute** intermediate-level Linux commands for file processing, text manipulation, user management, and basic networking operations.
4. **Utilize** advanced Linux commands to manage system processes, monitor performance, handle storage devices, and perform system-level administration.
5. **Develop** and **execute** Bash shell scripts for task automation, using control structures, loops, functions, and job scheduling tools (e.g. cron).

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 possible, it helps improve the students' understanding.

MODULE – I	**- 7 hours**
Basic Linux Commands: Understanding Linux Concepts, application, Unix vs Linux vsWindows**;**System utility commands (**date, uptime, hostname, which, cal, bc,** etc.), system maintenance commands (**shutdown, reboot, halt, init**, etc.)**;** Download, Install, and Configure Linux Operating System: Oracle Virtual Box, **Ubuntu** Linux Installation (**mandatory**), different ways to installing Linux**;**Introduction to File System, File system structure description, File system navigation commands, File System Paths, Directory listing overview**;**Creating Files and Directories, Finding Files and Directories command (**find, locate, pipe**)**;** file permissions (**chmod**), file ownership (**chown, chgrp**), getting help (**man, whatis**, etc.), adding text to file, standard output to a file (tee command)**;** **Textbook –T1-> Chapter-11 and Chapter-32; T2-> Chapter8, Chapter10, Chapter11, Chapter14, Chapter15;**	
MODULE– II	**- 8 hours**
Intermediate Linux Commands: Filters and Text Processing Commands **(cut, sort, grep, awk, uniq, wc)**; Compare Files (**diff, cmp**, etc.), Compress and un-compress files or directories (**tar, gzip, gunzip**)**;**Truncate file size (**truncate**), Combining and Splitting Files (**cat and split**)**,** mount and unmount file commands (e.g., **sudo mount /dev/sdX1 /mnt,sudoumount /mnt,sudoumount -f /mnt)**; Linux vs. Windows commands, Linux File Editors (**vi text editor**), user account management**;**recover root password (single user mode), switch users and sudo access, monitor users**;**talkingto users (**users, wall, write**); networking commands (**ifconfig, ip, ping, traceroute, etc.**),connect Linux VM via Putty**;** **Textbook – T1->Chapter4.4, Chapter4.5, Chapter-5; T2->Chapter-15**	
MODULE– III	**- 7 hours**
Advanced Linux Command: Processes and schedules (**systems, ps, top, kill, crontab, at,** etc.), System Monitoring Commands (**top, df, dmesg, iostat 1, netstat, free, etc.)**; System logs monitor (/var/log), Finding System Information (**uname, cat /etc./redhat-release, cat /etc./*rel*, dmidecode**, etc.), Terminal Commands (**clear, exit, script, etc.)**; storage commands: disk partition (**df, fdisk,** etc.), add disk and create standard partition, Logical Volume Management (**LVM**)**;**add disk and create LVM partition, Extend disk using LVM, Adding swap space, file system check (**fsck and xfsrepair**), system backup (**dd command only**)**.** **Textbook--T1->Chapter-2, Chapter-11; T2->Chapter-16, Chapter-18**	
MODULE – IV I	**- 8 hours**

Shell Programming Basics: Introduction to **bash shell scripting**, **hello world printing**, comment**;** variables, constant, sourcing a script, troubleshooting and debugging a script**;** syntax, script, variables, data types, operators**; if...else, loops: for, do while, switch; functions; arrays, schedule (cron);**Hello World Script, Print Current Date and Time, **Read Input from User;** Simple Calculator (Addition, Subtraction), file creation and deletion, check if a file exists**;** If-Else Example (**Even or Odd Number**), **Loop Through a List of Files;** Countdown Timer (**using while loop**), Display Disk Usage (**using df and du**), display System Uptime (using uptime).

Textbook-- T1->Chapter23.9,23.10, Chapter24.2,24.3; T2->Chapter-24 and 27

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Given the USB sticks with Ubuntu 24.04 LTS ISO files, student can **build** and configure Linux operating system-based host machine. Same operation can be **solved** using the Oracle VirtualBox and Ubuntu 24.04 LTS ISO files.	**L3**
CO2	Given the Linux Operating System andthe basic linux commands whoami, cd, mkdir, rm, adduser, pid, etc. student can **examine** the basic files and directory structures, user's credentials, network and system parameters.	**L4**
CO3	Given the Linux Operating System, basic linux commands**,** and bash shell platform student can **develop** basic bash scripts to automate system tasks.	**L3**
CO4	Given the Linux operating System, advanced linux commands (specifically, top, df, dmesg, netstat, free, etc.) and bash shell platform student can **design** data processing, network and system monitoring, user and process management tools effectively.	**L3**

Text Books

1.Author: Paul Cobbaut , **Title:** *Linux Fundamentals* , **Publisher:** linux-training.be, 2015 (CEST)

2. Author: William E. Shotts, Jr. **Title:** *The Linux Command Line* **Edition:** 2nd Edition , No Starch Press , 2019

E-Resources:

- https://www.tutorialspoint.com/unix/shell_scripting.htm • https://linuxjourney.com/.
- https://www.geeksforgeeks.org/introduction-linux-shell-shell-scripting/

- https://linuxcommand.org/tlcl.php
- https://nptel.ac.in/courses/117106113
- https://www.edx.org/learn/linux/the-linux-foundation-introduction-to-linux
- GNU Bash Manual
- The Linux Documentation Project (TLDP)
- OverTheWire – Bandit (Linux WarGame)
- Codecademy – Learn the Command Line
- LearnShell – Interactive Shell Tutorials

Open Ended Lab Problems:

- Create a script that takes a single argument (a filename) and prints its contents to the screen.
- Write a script that asks the user for their name and greets them.
- Create a script that lists all files in the current directory.
- Write a script that checks if a file exists and prints a message indicating whether it does or not.
- Write a script that takes multiple arguments (filenames) and concatenates their contents into a single output file.
- Create a script that searches for a specific string in all files in the current directory and prints the filenames of those that contain the string.
- Write a script that sorts a list of numbers in ascending order and prints the result.
- Create a script that monitors a directory for new files and sends an email notification when a new file is detected.
- Write a script that backs up a directory by compressing its contents into a tarball.
- Write a script that implements a simple calculator, allowing users to perform arithmetic operations (+, -, *, /) on two numbers.
- Create a script that analyzes system logs to detect potential security threats and sends an alert to the administrator.
- Write a script that automates a daily backup process, including rotating old backups and sending notifications.
- Create a script that monitors system resources (CPU, memory, disk usage) and sends alerts when thresholds are exceeded.
- Write a script that implements a simple chat server, allowing multiple users to communicate with each other.
- Write a script that solves the " Tower of Hanoi" problem for a given number of disks.
- Create a script that generates a random password of a specified length and complexity.
- Write a script that implements a simple game (e.g., Tic-Tac-Toe, Hangman) in the terminal.
- Create a script that analyzes a large dataset and generates statistics (e.g., mean, median, mode).
- Write a script that automates a complex workflow, involving multiple steps and conditional logic.

DIFFERENTIAL EQUATIONS & OPTIMIZATION

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

SEMESTER - IV

Course Code: 25CY2401	**Credits: 03**
Hours / Week: 03	**Total Hours: 45**
L–T–P–J:3–0–0–0	**CIE+SE:60+40 Marks**

Course Vision:

To develop strong mathematical and computational foundations in differential equations, 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. Apply Laplace transforms and analytical methods to solve differential equations arising in engineering applications.
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.

Teaching-Learning Process (General Instructions)

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

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

MODULE – I	**09 Hours**
Differential Equations Topics: Introduction to first-order ordinary differential equations (ODE), linear higher order differential equations with constant coefficients, homogeneous and non-homogeneous equations, Existence and uniqueness of solutions for ordinary differential equations, Method of variation of parameters, Introduction of Laplace transforms and its properties, Solving ODE using Laplace transform. Applications: Control systems engineering, Electrical circuit analysis, Mechanical vibration systems, Population dynamics modelling	
MODULE – II	**09 Hours**
Numerical Methods for ODE and PDE Topics: Solving initial value problems using Taylor series method, Euler's method and modified Euler method, Runge-Kutta methods, multistep methods. Higher-order equations and systems. Stability analysis. Solving boundary value problems using shooting methods, difference methods and variational methods. Introduction to numerical solutions for partial differential equations. Applications: Weather prediction models, Mechanical system simulations, Control system modelling, Heat transfer and diffusion problems.	
MODULE – III	**09 Hours**
Linear Programming Topics: Linear Programming: Linear programming, standard form of linear programming, geometry of linear programming problems, solution of a system of linear simultaneous equations, pivotal production of general systems of equations, graphical method, simplex algorithms, revised simplex methods, duality in linear programming. Introduction to assignment and transportation problem. Applications: Resource allocation and production planning, Transportation and logistics optimization, Supply chain management, Financial portfolio optimization, Scheduling and assignment problems	
MODULE – IV	**09 Hours**
Introduction to Optimization Topics: Introduction to classical optimization, engineering applications of optimization, classification of optimization problems, single variable optimization, multivariable optimization without constraints, multivariable optimization with equality and inequality constraints, Kuhn-Tucker conditions. Applications: Resource allocation problems, Engineering design optimization, Production	

planning, Logistics and supply chain optimization	
MODULE – V	**09 Hours**
Nonlinear Optimization Topics: Unconstrained nonlinear optimization, univariate method, gradient of a function, gradient descent method, Fletcher-Reeves method, constrained nonlinear optimization, characteristics of a constrained optimization problem, cutting plane method, interior and exterior penalty function methods. Applications: Neural network training, Portfolio optimization, Control system optimization, Engineering system design	

Authentic Intelligence:

1. Understanding limitations of dynamic models
2. Evaluating numerical errors in computational models
3. Ethical considerations in automated optimization systems
4. Ethical decision-making in optimization problems
5. Understanding biases introduced by optimization algorithms

AI Integration:

1. Dynamic system modelling in AI robotics
2. Simulation of dynamic systems in robotics
3. Decision optimization in intelligent systems
4. Linear programming used in decision-making systems
5. Gradient descent optimization algorithms for training neural networks and deep learning

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Solve ordinary differential equations and apply Laplace transforms for modelling dynamic systems used in AI,signal processing, and control systems.	L3
2	Implement numerical methods for solving ordinary and partial differential equations arising in engineering and scientific models.	L3
3	Develop and solve linear programming models for real-world problems	L3
4	Formulate and solve classical optimization problems including single and multivariable optimization with constraints.	L4
5	Apply nonlinear optimization techniques such as gradient descent, Fletcher–Reeves method, and penalty methods to engineering applications.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. Boyce W. E. and DiPrima R. C., Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
3. Grewal B.S., Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2010.
4. Taha, Hamdy A. *"Operations Research: An Introduction"*, 10th, Pearson Education, 2017.
5. Rao, Singiresu S. *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.

REFERENCE BOOKS:

1. Bird, John. Higher Engineering Mathematics. 6th ed., Elsevier, 2017.

2. Nayak, Sukanta. *Fundamentals of optimization techniques with algorithms*. Academic Press, 2020.
3. Arora, R. K. Optimization: Algorithms and Applications. 1st ed., Chapman and Hall/CRC, 2015.
4. Deb, Kalyanmoy. Optimization for Engineering Design: Algorithms and Examples. 2nd ed., PHI Learning Pvt. Ltd., 2012.
5. Nocedal, Jorge, and Stephen J. Wright. *Numerical Optimization*. 2nd ed., Springer, 2006.

OPEN ENDED LAB EXERCISES

S. No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I: Exercises 1–3: Differential Equations and Transform Methods (03 Hours)					
1	**Differential Equations (Basic)**	**Solve first-order and higher-order differential equations computationally**	**MATLAB/PYTHON**	**1 h**	**CO1**
2	**Laplace Transform**	**Use Laplace transforms to solve initial value problems.**	**MATLAB/PYTHON**	**1 h**	**CO1**
3	**Case Study of Dynamic Modelling**	**Simulate time-series data using differential equation models.**	**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)					
10	Formulation of LPP	Formulate and solve linear programming problems (LPP) and implement graphical method	MATLAB/PYTHON	1 h	CO4
11	Simple 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	ImplementFletcher-Reevesoptimization 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**

CLOUD SECURITY

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

SEMESTER – IV

Subject Code : 25CY2402	**Credits**:3
Hours / Week :03 Hours	**Total Hours** :45Hours
L–T–P–S : 3–0–0–0	

Course Learning Objectives:

This course will enable students to:

1. Master cloud basics, architecture, and deployment models to improve how your organization manages its IT infrastructure and operations.
2. Master the essential principles and practices of securing cloud-based systems to safeguard data confidentiality, integrity, and availability.
3. Understand and address security risks in cloud computing environments.
4. Understand how to effectively secure cloud computing environments through principles of architecture, trusted methodologies, identity management, and access control.
5. Demonstrate cloud computing life cycle issues, standards, and security measures, and to develop incident response plans.

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 the traditional lecture method but a different *type of teaching method* 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. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	**09 Hours**

Introduction & Cloud Computing Architecture: Cloud computing at a glance: Defining a cloud, A closer look, The cloud computing reference model, Characteristics and benefits Historical developments: Distributed systems, Virtualization, Web 2.0, Service oriented computing, Utility-oriented computing Building cloud computing environments: Application development, Infrastructure and system development, Computing platforms and technologies. The cloud reference model: Architecture, Infrastructure-and hardware-as-a securityservice, Platform as a service, Software as a service .Types of clouds: Public, Private, Hybrid, Open Challenges: Cloud definition, Cloud interoperability and standards, Scalability and fault tolerance, Security, trust, and privacy, Organizational aspects.

UNIT – II	10 Hours
Virtualization: Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques - Execution Virtualization, Other types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples – Xen, VMware, Microsoft Hyper-V Cloud Computing Software Security Fundamentals: Cloud Information Security Objectives-Confidentiality, Integrity, and Availability, Confidentiality, Integrity, Availability, Cloud Security Services-Authentication, Authorization, Auditing, Accountability	
UNIT – III	**09 Hours**
Secure Cloud Software Requirement-Secure Development Practices, Approaches to Cloud Software Requirements Engineering, Cloud Security Policy Implementation, NIST 33 Security Principles, *Secure Cloud Software Testing*-Testing for Security Quality Assurance, Cloud Penetration Testing, Regression Testing. **Cloud Computing Risk Issues and Security Challenges**: The CIA Triad, Privacy and Compliance Risks, Threats to Infrastructure, Data, and Access Control, Cloud Service Provider Risks, Cloud Computing Security Challenges-Security Policy Implementation, Computer Security Incident Response Team (CSIRT).	
UNIT – IV	**09 Hours**
Cloud Computing Security Architecture: Introduction, ***Architectural Considerations***- General Issues, Trusted Cloud Computing, Secure Execution Environments and Communications, Microarchitectures, ***Identity Management and Access Control***-Identity Management, Access Control, Autonomic Security	
UNIT – V	**08 Hours**
Cloud Computing Life Cycle Issues: Standards, The Distributed Management Task Force (DMTF), The International Organization for Standardization (ISO), The European Telecommunications Standards, The Organization for the Advancement of Structured, Storage Networking Industry Association (SNIA), Open Grid Forum (OGF), The Open Web Application Security Project (OWASP), Incident Response.	

Course Outcomes:

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

1. **Analyze** cloud computing fundamentals and architecture to optimize resource utilization and enhance organizational efficiency.
2. **Implement** cloud security measures effectively, including authentication, authorization, and auditing, to ensure confidentiality, integrity, and availability of data in cloud computing environments.
3. **Identify** cloud computing risks and security challenges, applying the CIA triad and privacy regulations. **Evaluate** threats to infrastructure, data, and access control, cloud service providers.
4. **Evaluate** cloud security principles, design secure environments and communications, integrate robust identity management, and enforce access controls for autonomic security.
5. **Assess** and **Analyze** cloud computing life cycle issues, standards from DMTF, ISO, ETSI, SNIA, OGF, and OWASP, and develop incident response plans.

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Rajkumar Buyya, Christian Vecchiola, S Tamarai Selvi "Mastering Cloud Computing Foundations And Applications Programming" , McGraw Hill Education, 2016.
2. Ronald L. Krutz, Russell Dean Vines “Cloud Security A Comprehensive Guide to Secure Cloud Computing” Wiley Publishing. INC.
3. Kai Hwang, Geoffrey C Fox, Jack J Dongarra, "Distributed and Cloud Computing - From Parallel Processing to the Internet of Things", Morgan Kaufman Publishing, 2012.

REFERENCE BOOKS:

1.Sirisha Potluri et.al. “Cloud Security Techniques and Applications (Smart Computing Applications)”.
2.Chris Dotson, “Practical Cloud Security A Guide for Secure Design and Deployment”. O’Riley Publishing.

E-Resources:

1. National Institute of Standards and Technology (NIST) - Cloud Computing Security: https://www.nist.gov/topics/cloud-computing-security
2. OWASP Cloud Security Top 10: https://owasp.org/www-project-cloud-security-top-10/
3. SANS Institute - Cloud Security: https://www.sans.org/white-papers/cloud-security/
4. Coursera - "Cloud Security Basics" by Google Cloud: https://www.coursera.org/learn/cloud-security- basics
5. edX - "Cloud Computing Security" by University of Maryland: https://www.edx.org/professional-certificate/cloud-computing-security
6. Pluralsight - "Cloud Security" learning path: https://www.pluralsight.com/paths/cloud-security
7. Cybrary - "Cloud Security Fundamentals" course: https://www.cybrary.it/course/cloud-security-fundamentals/

Activity Based Learning (Suggested Activities in Class)

1. Cloud Security as a group Case study.
2. Collaborative Activity is minor project development with a team of 4 students.

EMBEDDED SYSTEM DESIGN

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

SEMESTER - IV

Course Code	:	25CY2403	**Credits**	:	04
Hours / Week	:	04	**Total Hours**	:	45+30
L–T–P–J	:	3–0–2–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

To develop the ability to design, analyze, and implement embedded systems by integrating hardware, software, and real-time concepts for solving engineering problems.

Course Objectives:

This course will enable students:

1. To understand the fundamentals and design methodology of embedded systems.
2. To learn ARM architecture and embedded programming concepts.
3. To interface sensors, actuators, and communication protocols.
4. To apply real-time operating systems (RTOS) in embedded applications.
5. To develop embedded systems with communication, security, and power management features.

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. Use lecture method along with demonstrations for hardware-software integration.
2. Encourage interactive learning through discussions and problem-solving.
3. Demonstrate real-time applications using development boards.
4. Promote group-based mini projects and hands-on learning.
5. Use simulation tools and IDE-based debugging.
6. Encourage students to design and implement practical embedded solutions.

MODULE – I	**09 Hours**

Introduction to Embedded Systems

Topics: Definition and characteristics of embedded systems, comparison with general-purpose systems, applications, classification, design metrics, processor types (GPP, DSP, FPGA, MCU), memory types, embedded system design flow, UML basics.

Applications: Consumer electronics, automotive systems, industrial automation, Connected devices.

AI Integration: Basic understanding of integrating embedded systems with AI-enabled applications. Use of sensor data for simple intelligent decision-making and introduction to edge and cloud-based processing.

Authentic Intelligence: Understanding constraints of embedded systems in implementing AI and awareness of reliability and real-world deployment challenges.

MODULE – II	**09 Hours**
ARM Architecture & Embedded Programming **Topics:** ARM Cortex-M architecture, registers, memory map, pipeline, interrupts, instruction set basics, assembly programming, embedded C programming, compiler optimization, development tools. **Applications:** Microcontroller-based control systems, firmware development. **AI Integration:** Use of embedded programming to process data for intelligent applications and introduction to AI-enabled firmware design. **Authentic Intelligence:** Awareness of efficiency, optimization, and limitations in implementing intelligent algorithms on microcontrollers.	
MODULE – III	**09 Hours**
Embedded Hardware & Interfacing **Topics:** GPIO, timers, PWM, ADC, DAC, UART, SPI, I2C, CAN basics, sensor interfacing, actuator interfacing, watchdog timer, RTC. **Applications:** Sensor-based monitoring systems, motor control systems. **AI Integration:** Use of sensor data acquisition and interfacing techniques for AI-based monitoring and control systems. **Authentic Intelligence:** Understanding challenges in real-time data accuracy and reliability for intelligent embedded applications.	
MODULE – IV	**09 Hours**
Real-Time Operating Systems (RTOS) **Topics:** OS basics, real-time systems, tasks, scheduling (RMS, EDF), synchronization (semaphores, mutex), inter-task communication, RTOS basics, RTOS selection. **Applications:** Real-time control systems, multitasking embedded devices. **AI Integration:** Introduction to real-time task management for intelligent systems and handling multiple processes in AI-based applications. **Authentic Intelligence:** Understanding timing constraints and reliability issues in real-time intelligent embedded systems.	
MODULE – V	**09 Hours**
Communication Protocols & Embedded Security **Topics:** Wired and wireless communication (Wi-Fi, BLE, ZigBee, LoRa), communication protocols (MQTT, HTTP, CoAP), cloud platforms, embedded security basics (encryption, authentication, secure boot), power management. **Applications:** Smart home systems, industrial automation, remote monitoring. **AI Integration:** Application of communication protocols and embedded systems for data-driven intelligent applications and remote monitoring using connected devices. **Authentic Intelligence:** Understanding challenges in secure communication, data reliability, and constraints in implementing intelligent features in embedded systems.	

Course Outcomes

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Describe the fundamentals, characteristics, and design methodology of embedded systems.	L2
2	Analyze ARM architecture and develop embedded programs using C/assembly language.	L3
3	Apply interfacing techniques for sensors, actuators, and communication protocols in embedded systems.	L3
4	Apply RTOS concepts such as task scheduling, synchronization, and inter-task communication in embedded applications	L3
5	Develop embedded systems with communication, security, and power management features Techniques for improving the model performance.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, 3rd Edition, Elsevier, 2013.
2. Raj Kamal, Embedded Systems: Architecture, Programming and Design, 3rd Edition, Tata McGraw-Hill, 2021.
3. Frank Vahid and Tony Givargis, Embedded System Design: A Unified Hardware/Software Introduction, Wiley, 2002.

REFERENCE BOOKS:

1. E. A. Lee and S. A. Seshia, "Introduction to Embedded Systems: A Cyber-Physical Systems Approach", 2nd Edition, MIT Press, 2017.
2. Patrick Schaumont, "A Practical Introduction to Hardware/Software Co-design", Springer, 2010.
3. Tammy Noergaard, Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers, 2nd Edition, Elsevier, 2012.

RECOMMENDED TOOLS

- Programming: Embedded C, C++
- Development Environments (IDE): STM32CubeIDE, Keil uVision, Arduino IDE, Platform
- Hardware Platforms: STM32 (ARM Cortex-M), Arduino (UNO/Nano), ESP32
- Debugging Tools: Serial Monitor, JTAG Debugger (optional)
- Communication Tools: PuTTY, Tera Term
- Networking Tools: MQTT Broker (Mosquitto), ThingSpeak / Blynk
- Simulation Tools (Optional): Proteus, Tinkercad

LAB EXERCISES- 30 Hours

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
MODULE I · Exercises 1–3 : Introduction to Embedded Systems (06 Hours)					
1	GPIO Programming	Implement LED blinking and switch interfacing using microcontroller	STM32 (CubeIDE)	2 h	CO1
2	Interrupt Handling	Develop interrupt-based button control for event-driven operation	STM32 (CubeIDE)	2 h	CO1
3	Timer Programming	Generate delay and periodic signals using hardware timers	STM32 (CubeIDE)	2 h	CO1
MODULE II · Exercises 4–7 : ARM Programming & Interfacing (08 Hours)					
4	PWM Generation	Generate PWM signals to control LED brightness and servo motor	STM32 (CubeIDE)	2 h	CO2
5	ADC Interfacing	Read analog data from sensor and display values	STM32 (CubeIDE)	2 h	CO2

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
6	UART Communication	Implement serial communication between microcontroller and PC	STM32 (CubeIDE), Tera Term / PuTTY	2 h	CO2
7	SPI Communication	Perform data transfer using SPI protocol with external device	STM32 (CubeIDE)	2 h	CO2
MODULE III · Exercises 8–10 : Embedded Communication & Control (06 Hours)					
8	I2C Communication	Interface I2C-based sensor and read data	STM32 (CubeIDE)	2 h	CO3
9	Motor Control	Control DC motor or servo motor using driver circuit	STM32 (CubeIDE)	2 h	CO3
10	CAN Communication (Demo)	Demonstrate basic CAN communication between nodes	STM32 (CubeIDE)	2 h	CO3
MODULE IV · Exercises 11–12 : RTOS (04 Hours)					
11	RTOS Task Creation	Create multiple tasks and implement scheduling using FreeRTOS	STM32 (CubeIDE with FreeRTOS)	2 h	CO4
12	RTOS Communication	Implement inter-task communication using queue/semaphore	STM32 (CubeIDE with FreeRTOS)	2 h	CO4
MODULE V · Exercises 13–15 : Communication Protocols & Embedded Security (06 Hours)					
13	Wi-Fi Communication	Implement wireless communication and transmit data using Wi-Fi module	ESP32 (Arduino IDE / PlatformIO)	2 h	CO5
14	Communication Protocol (MQTT/HTTP)	Implement data exchange using MQTT or HTTP protocol	ESP32, MQTT Broker / HTTP Server	2 h	CO5
15	Embedded Security Basics	Demonstrate authentication or basic secure data transmission in embedded system	STM32 / ESP32	2 h	CO5

INTRODUCTION TO DATA SCIENCE

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

SEMESTER - IV

Course Code	:	25CY2405	**Credits**	:	02
Hours / Week	:	2	**Total Hours**	:	30
L–T–P–J	:	2–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

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

Course Objectives:

This course will enable students:

1. Understand fundamental data science concepts and workflows
2. Apply statistical techniques and visualization for data analysis
3. Perform data cleaning and preprocessing
4. Use feature selection and dimensionality reduction methods
5. Gain exposure to scalable data analytics techniques

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**06 Hours**

Basic Concepts and Python Packages

Topics: Predictive Modelling, Introduction: Data Preparation, Importance of Data Preparation, Data Cleaning, Feature Selection, Data Transformation, Python Packages: NumPy, Matplotlib, Pandas, SciPy, Scikit-learn, DataFrames, Loading Machine Learning Data

Applications: Data preprocessing in real-world problems, Basic analytics workflows

AI Integration: Data preparation for AI systems

Authentic Intelligence: Understanding data quality and reliability

MODULE – II	**06 Hours**

Descriptive Statistics and Data Visualization

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

Applications: Data analysis and reporting, Visualization-driven decision making

AI Integration: Statistical modeling and visualization in AI

Authentic Intelligence: Interpreting statistical and visual insights correctly

MODULE – III	**06 Hours**

Server-Side Development with Node.js and Express.js

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

Applications: Data cleaning in industry datasets, Feature engineering workflows

AI Integration: Improving model performance through feature selection

Authentic Intelligence: Ensuring reliable and unbiased data

MODULE – IV	**06 Hours**

Databases for Web Applications

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

Applications: Data transformation pipelines, High-dimensional data analysis

AI Integration: Feature optimization for machine learning

Authentic Intelligence: Reducing complexity while preserving information

MODULE – V	**06 Hours**

Web Security, Performance, and Deployment

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, Case study-population-scale data analysis for epidemic control, Integration of multiple datasets, Real-time data analysis and iterative model updates, Decision-making using large-scale data insights

Applications: Large-scale data analysis, Real-time decision-making systems

AI Integration: GPU-accelerated ML pipelines, High-performance AI systems

Authentic Intelligence: Efficient handling of large datasets, Scalable and reliable data-driven solutions

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain basic data science concepts and data pre-processing techniques	L2
2	Apply statistical methods and visualization techniques for data analysis	L3
3	Perform data cleaning and feature selection for efficient data preparation	L4
4	Apply data transformation and dimensionality reduction techniques	L3
5	Understand scalable data analytics concepts using modern tools	L2

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

REFERENCE BOOKS:

1. Joel Grus – Data Science from Scratch
2. Wes McKinney – Python for Data Analysis
3. Jake VanderPlas – Python Data Science Handbook
4. Online documentation (Pandas, Scikit-learn, Matplotlib)
5. DLILink:https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-DS-03+V1#:~:text=The%20course%20prerequisites%20include:%20*%20Basic%20understanding,the%20NVIDIA%20Deep%20Learning%20Institute%20at%20www.nvidia.com/dli

AI Essentials for Cyber Security

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

SEMESTER – IV

Subject Code	: 25CY2406	Credits : 04
Hours / Week	: 03 Hours	Total: 45 (Th)+30(P)Hours
L–T–P–J	: 3–0–2–0	

Course Learning Objectives:

This course will enable students to:

1. Understand the fundamentals of Artificial Intelligence, Machine Learning, and their applications in cyber security.
2. Analyze cyber threats using intelligent data-driven techniques.
3. Apply supervised, unsupervised, and ensemble learning models for cyber security problems.
4. Design AI-enabled solutions for anomaly detection, phishing detection, and network defense.
5. Explore Generative AI, Large Language Models (LLMs), prompt engineering, and AI-assisted threat intelligence.

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 the traditional lecture method but a different *type of teaching method* 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. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I Introduction to AI-ML	**09 Hours**
Introduction to Artificial Intelligence and Machine Learning, Evolution of AI in Cyber Security, Cyber threat landscape and intelligent defense systems, Types of Machine Learning Algorithms- Supervised Learning,Unsupervised Learning,Reinforcement Learning; Data preprocessing and feature engineering,ML pipeline architecture- Training, validation, and testing concepts, Overfitting, Underfitting, Bias, Variance, Classification and prediction techniques, Introduction to cyber security datasets, Challenges in AI-driven cyber defense	
Applications- Malware classification,Intrusion detection systems,Intelligent firewall systems,AI-assisted threat prioritization,Security log analysis AI Integration- Using Python libraries for ML model creation,Introduction to Scikit-learn and Jupyter Notebook,AI-based cyber analytics dashboards Authentic Intelligence Integration-Responsible AI usage in cyber security,Bias and fairness in security datasets,Human-in-the-loop cyber defense	
UNIT – II Time Series Analysis	**12 Hours**
Introduction to Time Series Forecasting, Components of Time Series Data-Trend, Seasonality, Noise, Cyclic behaviour; Time Series decomposition techniques, Cyber security event forecasting, DDoS attack prediction using ML, Classification and Regression Algorithms- Binary and Multi-Class Classification, Regression - Linear Regression and Nonlinear Regression, Logistic Regression, Decision Trees (ID3, CART) and Random Forest, Regularization - L1 Regularization, L2 Regularization Ensemble Learning Methods, Voting and Stacking techniques, Bagging Algorithms, Understanding Out-Of-Bag Score, Boosting: AdaBoost, Gradient Boosting Machines (GBM), XGBoost, CatBoost.	
Applications- DDoS attack forecasting, Network traffic prediction, User behavior analytics, Security event forecasting AI Integration- Ensemble-based cyber-attack prediction, AI-driven anomaly trend analysis Authentic Intelligence Integration- Trustworthy forecasting systems, Ethical use of behavioral analytics, Minimizing false positives in intelligent defense systems	
UNIT – III Malicious URL Detection	**08 Hours**
Introduction to malicious URLs and phishing attacks, Types of URL-based cyber attacks, Feature extraction from URLs, Heuristic-based malicious page detection, Machine Learning approaches for URL classification- Logistic Regression for phishing detection, Support Vector Machine (SVM), Browser security analytics, AI-enabled phishing detection systems	
Applications- Phishing website detection, Fake domain identification, Browser security extensions AI Integration- AI-assisted phishing detection platforms, Real-time malicious URL classifiers	

Authentic Intelligence Integration- Ethical use of user browsing data, Privacy-preserving web security analytics, Human verification in AI-generated alerts	
UNIT – IV CAPTCHA **Analysis, and Network Anomaly Detection**	**08 Hours**
Fundamentals of CAPTCHA systems, Characteristics and types of CAPTCHA, AI techniques used to crack CAPTCHA, Defensive CAPTCHA mechanisms, Email Spoofing attacks, Spam detection using Machine Learning, Clustering techniques in cyber security- K-Means clustering algorithm; Network anomaly detection using clustering, Insider threat detection basics. Machine Learning Optimization Algorithms: Gradient Descent Techniques, Bayesian Optimization, Overfitting, Underfitting, Optimal Fit, Bias, Variance, Regularization - L1 Regularization, L2 Regularization, Elastic Net Regularization.	
Applications- Spam email filtering systems, Email fraud detection, Network anomaly monitoring, Insider threat detection AI Integration- AI-powered spam filters, Intelligent network monitoring systems Authentic Intelligence Integration- Ethical AI in email surveillance, Trustworthy anomaly detection systems, Responsible monitoring and privacy preservation	
UNIT – V Intelligent **Threat Analysis & LLMs**	**07 Hours**
Context-based malicious event detection ,Event correlation and behavioral analysis ,Rule-based vs AI-based threat detection systems ,Security incident classification using Decision Trees ,Threat intelligence and attack pattern analysis ,Introduction to Generative AI in cyber security , Fundamentals of Large Language Models (LLMs) ,Transformer architecture basics ,Tokenization and embeddings , Prompt Engineering fundamentals ,Types of prompts:Zero-shot prompting,Few-shot prompting ,Chain-of-thought prompting; AI-powered threat analysis and cyber intelligence, Automated incident response using AI	
Applications- Intelligent malware detection system, Automated cyber incident summarization, Context-aware intrusion detection AI Integration- Using LLMs for threat intelligence generation , Prompt engineering for cyber investigation workflows Authentic Intelligence Integration- Human-in-the-loop security decision systems , Responsible use of Generative AI in cyber defense	

Course Outcomes

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Illustrate the fundamental concepts of Artificial Intelligence, Machine Learning, and cyber security analytics	**L2**
2	Apply supervised, unsupervised, and ensemble learning algorithms suitable for cyber security problems	**L3**
3	Identify malicious URLs, phishing attacks, spam emails, and network anomalies using intelligent techniques	**L3**
4	Illustrate predictive threat analysis and cyber defense automation	**L2**
5	Explain LLMs, and Authentic Intelligence principles for trustworthy and ethical cyber security systems.	**L2**

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. A. Hands-on Machine Learning for Cyber Security by Soma Halder, ISBN139781788992282
2. Machine Learning – Tom M. Mitchell, McGraw-Hill, 1st Edition, 1997
3. 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.

RECOMMENDED TOOLS

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

LAB EXERCISES – 30 Hours

S.No	Exercise Title	Aim / Objective	Tools	CO
1	Data Analysis and Visualization	Perform Exploratory Data Analysis and Visualize Features using the suitable Dataset (e.g: Iris Dataset)	Google Colab/ Jupyter	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	CO1
3	Split Dataset, cross validation	Split Dataset into Training and Testing Sets and Apply Cross-Validation.	Google Colab/ Jupyter	CO1
4	Linear Regression	Implement Linear Regression to Predict House Prices using the Housing Price Dataset.	Google Colab/ Jupyter	CO2
5	Logistic Regression	Build a Logistic Regression Model to Perform Classification	Google Colab/ Jupyter	CO2
6	K-Nearest Neighbors (KNN)	Apply the K-Nearest Neighbors (KNN) Algorithm to perform classification on suitable dataset (e.g Classify Flower Species using the Iris Dataset)	Google Colab/ Jupyter	CO2

S.No	Exercise Title	Aim / Objective	Tools	CO
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	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	CO2
9	K-Means Clustering	Apply K-Means Clustering to Perform Customer Segmentation. (e.g using the Mall Customer Segmentation Dataset)	Google Colab/ Jupyter	CO3
10	Principal Component Analysis	Apply Principal Component Analysis (PCA) to Reduce Feature Dimensions. (e.g: using the Wine Dataset)	Google Colab/ Jupyter	CO3
11	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	CO5
12	Machine Learning Model to Solve a Real-World Engineering	Design and Implement a Machine Learning Model to Solve a Real-World Cyber Security Problem using an Open Engineering Dataset	Google Colab/ Jupyter	CO5

WEB TECHNOLOGIES

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

SEMESTER - IV

Course Code	:	25CY2408	**Credits**	:	02
Hours / Week	:	2	**Total Hours**	:	30
L–T–P–J	:	2–0–0–0	**CIE**	:	100 Marks

Course Vision:

To equip students with industry-relevant skills in designing, developing, and deploying modern web applications by integrating foundational web standards, client-side and server-side technologies, database connectivity, security practices, and cloud deployment - preparing them to contribute effectively to real-world software engineering projects and entrepreneurial ventures in the digital economy.

Course Objectives:

This course will enable students:

1. Understand the architecture of the World Wide Web, HTTP/HTTPS protocols, and apply HTML5 and CSS3 to build semantically structured, responsive web pages.
2. Develop dynamic, interactive client-side applications using JavaScript, its modern ES6+ features, and the DOM API.
3. Design and implement server-side RESTful web services and back-end logic using Node.js and Express.js.
4. Integrate SQL and NoSQL (MongoDB) databases with web applications for persistent and efficient data management.

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**07 Hours**

Web Fundamentals, HTML5 and CSS3

Topics: Web Fundamentals: Client–Server Architecture, Web Browsers, URL and DNS Basics, HTTP & HTTPS: Request–Response Lifecycle, HTTP Methods, Status Codes, HTML5: Document Structure, Semantic Elements, Forms, Multimedia, CSS3: Selectors, Box Model, Flexbox, CSS Grid, Responsive Web Design.

Applications: Static and responsive multi-page web portals, Portfolio and resume websites, Online form interfaces for data collection, Accessible government and educational websites

AI Integration: AI tools (GitHub Copilot, ChatGPT) for accelerated HTML/CSS scaffolding and debugging, Automated accessibility testing using AI-based tools, AI-powered responsive design suggestions (Adobe Firefly, Framer AI), Code linting and best-practice enforcement via AI-integrated IDEs.

Authentic Intelligence: Understanding that AI-generated HTML must be manually validated for semantic correctness and accessibility, Critical evaluation of auto-generated CSS for performance and maintainability

NVIDIA DLI Courses (Relevant to this Module):

1. Getting Started with Deep Learning (8 hrs) | Path: Deep Learning — Foundational Deep Learning for AI-Driven Web Applications

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-FX-01+V3

2. Building Real-World AI Applications with NVIDIA NIM (2 hrs) | Path: Generative AI — Deploying LLM Microservices via REST APIs

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-NIM-01+V1

MODULE – II	**08 Hours**

JavaScript and Client-Side Development

Topics: JavaScript Fundamentals: Variables, Data Types, Scope, Functions, Conditionals and Loops, ES6 Features: Arrow Functions, Template Literals, Destructuring, Spread Operator, DOM Manipulation, Event Handling, Asynchronous JavaScript: Promises and Async/Await, Fetch API, Client-Side Storage: localStorage, sessionStorage, Cookies.

Applications: Dynamic form validation and real-time feedback, Single-page interaction without page reloads, Browser-based games and interactive dashboards, Client-side data filtering and search

AI Integration: AI-assisted JavaScript debugging (GitHub Copilot, Tabnine), AI-powered code review for detecting runtime errors

Authentic Intelligence: Responsible use of asynchronous patterns to prevent UI blocking and poor UX, Security implications of storing sensitive data in localStorage

MODULE – III	**08 Hours**

Server-Side Development with Node.js and Express.js

Topics: Node.js Fundamentals: Event-Driven Architecture, npm and package.json, Built-in Modules, Express.js: Application Setup, Routing, Request and Response Handling, Middleware, RESTful API Development, Template Engine Basics (EJS), Authentication and Authorization, Password Hashing with bcrypt.

Applications: RESTful back-end APIs for mobile and web frontends, Authentication services for SaaS platforms, File upload and download services

AI Integration: Integrating OpenAI / Anthropic APIs via Node.js HTTP clients, AI-assisted, API documentation generation (Swagger + AI), Intelligent rate limiting and anomaly detection using AI middleware

Authentic Intelligence: Secure storage of credentials - avoiding hardcoded secrets (use .env and dotenv), JWT expiry and refresh token strategies to balance security and UX

NVIDIA DLI Free Courses (Relevant to this Module):

1. **Building Conversational AI Applications** (8 hrs) | Path: Conversational AI — Integrating LLM APIs via Node.js REST Back-Ends

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-17+V1

MODULE – IV	**07 Hours**

Databases for Web Applications

Topics: Relational Databases: SQL Basics, DDL and DML Operations, Keys and JOINs, Preventing SQL Injection, Connecting Databases with Node.js, NoSQL and MongoDB Basics, MongoDB CRUD Operations, Query Operators, Indexing and Aggregation Basics, Mongoose ODM: Schemas, Models, Validation, and Populate.

Applications: User profile and credential management systems, Real-time analytics dashboards backed by MongoDB aggregation, Mechanical vibration systems, Healthcare patient record management systems

AI Integration: AI-assisted query optimization and schema suggestions, Intelligent database indexing recommendations

Authentic Intelligence: Data privacy and GDPR compliance - responsible storage of personal data, Preventing SQL injection through parameterized queries and ORMs

NVIDIA DLI Courses (Relevant to this Module):

1. **Introduction to AI in the Data Center** (1 hrs) | Path: Infrastructure — AI-Ready Database Architecture & Cloud Storage Design

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-FX-01+V1

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the fundamentals of web technologies, HTTP/HTTPS, and create structured web pages using HTML5 and CSS3.	L2
2	Develop responsive and interactive front-end web applications using JavaScript, DOM manipulation, event handling, and asynchronous programming.	L3
3	Build server-side web applications and RESTful APIs using Node.js, Express.js, middleware, and authentication mechanisms.	L3
4	Apply SQL and MongoDB database operations for designing and integrating data-driven web applications.	L3

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. HTML & CSS: Design and Build Web Sites by Jon Duckett
2. JavaScript: The Definitive Guide, Seventh Edition by David Flanagan
3. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett
4. Web Development with Node and Express, 2nd Edition by Ethan Brown

5. MongoDB: The Definitive Guide, 3rd Edition by Shannon Bradshaw, Eoin Brazil, Kristina Chodorow

REFERENCE BOOKS:

1. Douglas Crockford, JavaScript: The Good Parts, O'Reilly Media, 1st Edition
2. Thomas A. Powell, HTML & CSS: The Complete Reference, McGraw-Hill

RECOMMENDED TOOLS

- Frontend: HTML5, CSS3, JavaScript (ES6+), React.js (optional but recommended),Bootstrap / Tailwind CSS
- Backend: Node.js, Express.js, Postman / Thunder Client
- Databases: MongoDB (Atlas), MySQL / PostgreSQL
- Dev Tools: Visual Studio Code, Git & GitHub
- Deployment: Vercel / Netlify (frontend), Render / AWS (backend)
- AI Assistants: GitHub Copilot, ChatGPT, Claude

DevOps

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

SEMESTER - IV

Course Code	:	25CY2409	**Credits**	:	02
Hours / Week	:	2	**Total Hours**	:	30
L–T–P–J	:	2–0–0–0	**CIE**	:	100 Marks

Course Vision:

To build industry-ready software engineering professionals who can design, automate, and operate the complete software delivery lifecycle - from source-code management and automated testing through containerisation, orchestration, cloud deployment, and site reliability engineering - by internalising an automation-first, security-by-design, and blameless-culture mindset that accelerates delivery while ensuring system resilience, observability, and responsible stewardship of infrastructure.

Course Objectives:

This course will enable students:

1. To understand the DevOps philosophy, cultural principles (CALMS framework, Three Ways), and the complete software delivery lifecycle from commit to production.
2. To apply Linux system administration, Bash scripting, and advanced Git version-control workflows essential for automating and managing DevOps processes.
3. To design and implement automated CI/CD pipelines using GitHub Actions and Jenkins, integrating code-quality gates, security scanning, artifact management, and Infrastructure as Code with Terraform and Ansible.
4. To containerise applications using Docker and orchestrate production workloads on Kubernetes, configuring services, ingress, auto-scaling, and Helm-based package management.

Teaching-Learning Process (General Instructions)

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

1. **Lecture method** along with traditional lecture method, different types 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 role-playing.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**08 Hours**

DevOps Foundations, Linux System Administration & Advanced Git

Topics: DevOps Fundamentals (Philosophy, CALMS, Three Ways, DevOps vs Agile vs SRE, DORA Metrics), DevOps Lifecycle & Toolchain (Plan to Monitor, Tool Ecosystem, Value Stream Mapping), Linux Basics for DevOps (Filesystem, Commands, Permissions, Processes, Package Management, SSH & File Transfer), Bash Scripting (Conditionals, Loops, Functions, Error Handling, Scheduling), Networking Basics (OSI/TCP-IP, DNS, HTTP/HTTPS, Ports, Network Tools, Firewall & SSH), Git & Version Control (Git Internals, Branching Strategies, History Management, Collaboration & Code Review, Versioning).

Applications: Automated server initialisation Bash script (EC2 user-data style) — installs packages, creates users, configures SSH, Log-parsing pipeline using grep + awk + sed to extract error rates from Nginx access logs, Scheduled daily backup automation with cron: database dump → gzip → upload to cloud object storage

AI Integration: GitHub Copilot for Bash script generation, documentation, and unit test scaffolding, Natural language to shell command translation using LLMs (Claude CLI, ChatGPT) for DevOps automation, LLM-generated runbooks and post-mortem templates from incident alert context

Authentic Intelligence: Never blindly execute AI-generated scripts - mandatory review for destructive operations (rm -rf, DROP TABLE) and privilege escalation, Ethical responsibility: a poorly tested cron job can cause a production outage affecting thousands of users

NVIDIA DLI Courses (Relevant to this Module):

1. Fundamentals of Linux for DevSecOps and Infrastructure Automation (Free | ~4 hrs) | Path: Infrastructure & DevOps — Foundations

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-SY-01+V1

MODULE – II	**08 Hours**

Containerisation with Docker & Container Registry Management

Topics: Containers & Virtualization Basics (Containers vs VMs, Namespaces, cgroups, Union Filesystems, OCI), Docker Architecture (Docker Daemon, CLI, containers, runc, Client–Server Model), Dockerfile & Image Building (Core Instructions, Multi-stage Builds, Layer Caching, Base Images, Security Best Practices), Docker CLI & Runtime (Build, Run, Exec, Logs, Resource Limits, Restart Policies), Docker Networking & Storage (Network Types, DNS, Port Mapping, Volumes, Bind Mounts, Persistence), Docker Compose (Service Configuration, Dependencies, Environment Management, Multi-environment Setup), Container Registry & Image Management (Docker Hub/GHCR/ECR, Tagging Strategies, Multi-arch Builds, Image Scanning)

Applications: Multi-stage Dockerfile for a Python FastAPI service - builder stage (pip install) + distroless final image, reducing image size by over 80%, Containerising a legacy monolith by extracting it into a reproducible Docker image with environment-based configuration

AI Integration: GitHub Copilot for generating Dockerfiles, .dockerignore, and docker-compose.yml with context-aware multi-stage suggestions, AI-assisted Dockerfile linting and security hardening (Hadolint + Semgrep rules integrated with Git pre-commit hooks)

Authentic Intelligence: Never run containers as root in production — principle of least privilege reduces blast radius of container escapes, SHA digest pinning over mutable tags (latest) prevents silent supply-chain attacks from upstream image tampering

MODULE – III	**07 Hours**

CI/CD Pipelines, Infrastructure as Code & Configuration Management

Topics: CI/CD Fundamentals (CI vs CD, Pipeline Stages, Pipeline as Code, Shift-left Testing), GitHub Actions (Workflows, Triggers, Runners, Reusable Workflows, Secrets, Caching, Matrix Builds, Deployments), Jenkins (Master–Agent, Jenkinsfile, Pipeline Stages, Shared Libraries, Docker Integration), Code Quality & Security (Static Analysis, Pre-commit Hooks, Dependency Scanning, Quality Gates), Artifact Management (Artifacts, Repositories, Versioning & Promotion), Infrastructure as Code with Terraform (HCL, Providers, State Management, Lifecycle, Modules, Workspaces), Configuration Management with Ansible (Inventory, Playbooks, Roles, Variables, Templates, Vault, Idempotency), GitOps Basics (Principles, Pull vs Push CD, ArgoCD Overview)

Applications: Jenkins multi-branch pipeline for a Java Spring Boot service with SonarQube quality gate and Nexus artifact promotion, Ansible playbook for zero-downtime rolling deployment of a web application across 10 VMs with pre/post health checks

AI Integration: AI-generated Terraform modules and Ansible playbooks using GitHub Copilot and Amazon CodeWhisperer, AI-powered IaC security scanning using Checkov and tfsec with LLM-generated remediation explanations

Intelligent rate limiting and anomaly detection using AI middleware

Authentic Intelligence: Immutable infrastructure principle: resist the temptation to SSH into production and patch manually ('pets vs cattle') — every change must flow through a pipeline, Responsible use of CI secrets: masked variables can still leak through log interpolation — audit pipeline logs before making repositories public

NVIDIA DLI Free Courses (Relevant to this Module):

1. Building CI/CD Pipelines for Machine Learning and AI Workloads on GPU Infrastructure (4 hrs) | Path: MLOps & DevOps for AI — Production AI Pipeline Engineering

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-22+V1

MODULE – IV	**07 Hours**

Container Orchestration with Kubernetes & Observability Stack

Topics: Kubernetes Fundamentals (Architecture, Control Plane & Worker Nodes, Networking, CNI, Declarative vs Imperative), Core Kubernetes Objects (Pod, ReplicaSet, Deployment, StatefulSet, DaemonSet, Jobs, Namespaces), Services & Ingress (Service Types, Ingress, DNS & Service Discovery), Configuration & Storage (ConfigMaps, Secrets, Volumes, PV/PVC, StorageClasses), Resource Management & Scaling (Requests/Limits, Quotas, HPA, Autoscaling), Helm (Charts, Templates, Releases, Repositories, Configuration Overrides), Kubernetes Security (RBAC, Service Accounts, Pod Security, Network Policies, Secrets Management), Observability (Monitoring with Prometheus & Grafana, Logging with EFK, Distributed Tracing & OpenTelemetry)

Applications: Deploy a 3-tier microservices application on Kubernetes with Deployments, ClusterIP and LoadBalancer Services, Nginx Ingress with TLS, ConfigMaps for environment config, and HPA targeting 70% CPU, Full kube-prometheus-stack installation: Prometheus scraping custom application metrics, Grafana dashboards with SLO burn-rate panels, Alertmanager Slack integration

AI Integration: k8sGPT for natural-language Kubernetes cluster diagnostics — AI interpretation of CrashLoopBackOff, Pending pods, and resource quota breaches, AI-powered anomaly detection in Prometheus time-series metrics (Amazon DevOps Guru, Dynatrace Davis AI, Grafana ML)

Authentic Intelligence: CPU/memory limits are not optional in production - unbounded containers cause noisy-neighbour eviction of critical services, RBAC misconfiguration is the number-one Kubernetes security finding in cloud audits — least-privilege must be designed from day one, not retrofitted

NVIDIA DLI Courses (Relevant to this Module):

1. Deploying AI Workloads on Kubernetes with NVIDIA GPU Operator and Multi-Instance GPU (MIG) (8 hrs) | Path: Infrastructure & DevOps — GPU-Accelerated Kubernetes

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-16+V1

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the DevOps philosophy, CALMS framework, Three Ways, DORA metrics, and design a complete DevOps toolchain map for a given software delivery scenario; write Bash automation scripts for server provisioning and log analysis; implement advanced Git branching strategies for team collaboration.	L2
2	Build production-grade Docker images using multi-stage Dockerfiles with security best practices (non-root, distroless, digest pinning), orchestrate multi-container applications with Docker Compose, and manage container images in a registry with automated vulnerability scanning.	L3
3	Design and implement end-to-end CI/CD pipelines in GitHub Actions and Jenkins integrating automated tests, code quality gates, SAST/SCA security scans, and artifact promotion; provision cloud infrastructure using Terraform modules with remote state; automate configuration management using Ansible roles.	L6
4	Deploy and manage containerised workloads on Kubernetes using Deployments, StatefulSets, Services, Ingress, HPA, RBAC, and Helm charts; configure a full observability stack (Prometheus + Grafana + EFK) with meaningful SLI dashboards and alert routing.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

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

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

TEXT BOOKS:

1. Gene Kim, Jez Humble, Patrick Debois & John Willis - The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, IT Revolution Press, 2nd Edition, 2021
2. Nigel Poulton - Docker Deep Dive, Independently Published, 2023 Edition.

REFERENCE BOOKS:

1. Jez Humble & David Farley - Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010
2. Brendan Burns, Joe Beda, Kelsey Hightower & Lachlan Evenson - Kubernetes: Up and Running, O'Reilly Media, 3rd Edition, 2022.

RECOMMENDED TOOLS

- Development Environment: VS Code (with Remote-SSH, Docker, Kubernetes extensions) | JetBrains Fleet | Vim/Neovim for terminal workflows
- Version Control: Git 2.40+ | GitHub / GitLab | GitHub Desktop (beginners) | GitKraken (visual Git history)
- Containerisation: Docker Desktop (macOS/Windows) | Podman (rootless alternative, Linux) | Docker Buildx | Trivy / Grype (image scanning) | Hadolint (Dockerfile linting)
- Kubernetes: minikube or kind (local cluster) | kubectl | k9s (TUI cluster manager) | Helm 3 | Lens IDE | k8sGPT (AI diagnostics)
- CI/CD: GitHub Actions | Jenkins (via Docker: jenkins/jenkins:lts) | ArgoCD | Flux CD | Tekton (cloud-native alternative)
- AI Assistants: GitHub Copilot, ChatGPT, Claude