**ANNA UNIVERSITY, CHENNAI**

**NON- AUTONOMOUS COLLEGES**

**AFFILIATED TO ANNA UNIVERSITY**

**MASTER OF COMPUTER APPLICATIONS (2 YEARS)**

**REGULATIONS 2025**

## PROGRAMME OUTCOMES (POs):

| PO | Programme Outcomes |
|---|---|
| **PO1** | An ability to independently carry out research /investigation and development work to solve practical problems |
| **PO2** | An ability to write and present a substantial technical report/document. |
| **PO3** | Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program |

| **PROGRAMME SPECIFIC OUTCOMES:** |
|---|
| **PSO1:** Design and develop computer applications by applying programming, database, networking, and system software knowledge to meet real-world requirements. |
| **PSO2:** Apply modern computing technologies and frameworks, including AI, cloud, and web platforms, to build efficient, user-friendly, and secure applications. |

**ANNA UNIVERSITY, CHENNAI**

**POSTGRADUATE CURRICULUM (NON-AUTONOMOUS AFFILIATED INSTITUTIONS)**

**Programme:** Master of Computer Applications (2 Years) **Regulations:** 2025

**Abbreviations:**

**BS** – Basic Science (Mathematics)

**ES –** Engineering Science (Programme Core (**PC**), Programme Elective (**PE**))

**SD** – Skill Development

**SL** – Self Learning

**OE** – Open Elective

**L** – Laboratory Course

**T** – Theory

**LIT** – Laboratory Integrated Theory

**PW** – Project Work

**TCP –** Total Contact Period(s)

**Semester – I**

| S. No. | Course Code | Course Title | Type | Periods per week | | | TCP | Credits | Category |
|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | | | |
| 1. | MC25101 | Mathematical Foundation of Computer Applications | T | 3 | 1 | 0 | 4 | 4 | BS |
| 2. | MC25102 | Networking and Communication Systems | T | 3 | 0 | 0 | 3 | 3 | ES (PC) |
| 3. | MC25103 | Software Testing and Automation | T | 3 | 0 | 0 | 3 | 3 | ES (PC) |
| 4. | MC25104 | Data Structures and Algorithms using Python | LIT | 2 | 0 | 2 | 4 | 3 | ES (PC) |
| 5. | MC25105 | Data Exploration and Visualization | LIT | 2 | 0 | 2 | 4 | 3 | ES (PC) |
| 6. | MC25106 | Advanced Java Programming | LIT | 2 | 0 | 2 | 4 | 3 | ES (PC) |
| 7. | MC25107 | Technical Seminar | - | 0 | 0 | 2 | 2 | 1 | SD |
| **Total Credits** | | | | | | | **24** | **20** | |

**Semester – II**

| S. No. | Course Code | Course Title | Type | Periods per week | | | TCP | Credits | Category |
|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | | | |
| 1. | MC25201 | UI & UX Design | T | 3 | 0 | 0 | 3 | 3 | ES (PC) |
| 2. | --- | Programme Elective I | T | 3 | 0 | 0 | 3 | 3 | ES (PE) |
| 3. | MC25202 | Foundations of Data Science | LIT | 3 | 0 | 2 | 5 | 4 | ES (PC) |
| 4. | MC25203 | Full Stack Web Development | LIT | 3 | 0 | 4 | 7 | 5 | ES (PC) |
| 5. | MC25204 | Mobile Application Development | LIT | 3 | 0 | 4 | 7 | 5 | ES (PC) |
| 6. | MC25205 | Quantum Computing | T | 2 | 0 | 0 | 2 | 2 | ES (PC) |
| 7. | --- | Industry-Oriented Course I | - | 1 | 0 | 0 | 1 | 1 | SD |
| 8. | MC25206 | Industrial Training | - | - | - | - | - | 2 | SD |
| 9. | --- | Self-Learning Course | - | - | - | - | - | 1 | - |
| | | | | | | **Total Credits** | **28** | **26** | |

**Semester – III**

| S. No. | Course Code | Course Title | Type | Periods per week | | | TCP | Credits | Category |
|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | | | |
| 1. | --- | Programme Elective II | T | 3 | 0 | 0 | 3 | 3 | ES (PE) |
| 2. | --- | Programme Elective III | T | 3 | 0 | 0 | 3 | 3 | ES (PE) |
| 3. | --- | Programme Elective IV | T | 3 | 0 | 0 | 3 | 3 | ES (PE) |
| 4. | --- | Open Elective | T | 3 | 0 | 0 | 3 | 3 | - |
| 5. | MC25301 | Machine Learning Techniques | LIT | 3 | 0 | 2 | 5 | 4 | ES (PC) |
| 6. | -- | Industry-Oriented Course II | -- | 1 | 0 | 0 | 1 | 1 | SD |
| | | | | | | **Total Credits** | **18** | **17** | |

**Semester – IV**

| S. No. | Course Code | Course Title | Type | Periods per week | | | TCP | Credits | Category |
|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | | | |
| 1. | MC25401 | Project Work | - | 0 | 0 | 20 | 20 | 10 | SD |
| | | | | | | **Total Credits** | **20** | **10** | |

**Total Credits for the Programme:80**

**Programme Elective Courses (PEC)**

| S. NO. | Course Code | Course Title | Periods Per Week | | | Total Contact Periods | Credits |
|---|---|---|---|---|---|---|---|
| | | | L | T | P | | |
| 1. | MC25001 | Big Data Analytics | 3 | 0 | 0 | 3 | 3 |
| 2. | MC25002 | VIBE coding | 3 | 0 | 0 | 3 | 3 |
| 3. | MC25003 | Network Programming and Management | 3 | 0 | 0 | 3 | 3 |
| 4. | MC25004 | Software Project Management | 3 | 0 | 0 | 3 | 3 |
| 5. | MC25005 | E-Learning | 3 | 0 | 0 | 3 | 3 |
| 6. | MC25006 | Accounting and Financial Management | 3 | 0 | 0 | 3 | 3 |
| 7. | MC25007 | Digital and Mobile Forensics | 3 | 0 | 0 | 3 | 3 |
| 8. | MC25008 | Cryptocurrency and Blockchain Technologies | 3 | 0 | 0 | 3 | 3 |
| 9. | MC25009 | Game Programming | 3 | 0 | 0 | 3 | 3 |
| 10. | MC25010 | Entrepreneurship Development | 3 | 0 | 0 | 3 | 3 |
| 11. | MC25011 | Generative AI and Prompt Engineering | 3 | 0 | 0 | 3 | 3 |
| 12. | MC25012 | Wireless Sensor Networks and Body Area Network | 3 | 0 | 0 | 3 | 3 |
| 13. | MC25013 | Fog and Edge Computing | 3 | 0 | 0 | 3 | 3 |
| 14. | MC25014 | Internet of Things | 3 | 0 | 0 | 3 | 3 |
| 15. | MC25015 | Deep Learning Techniques | 3 | 0 | 2 | 5 | 4 |
| 16. | MC25016 | Artificial Intelligence | 3 | 0 | 2 | 5 | 4 |
| 17. | MC25017 | DevOps | 3 | 0 | 2 | 5 | 4 |
| 18. | MC25018 | Ethical Hacking and Penetration Testing | 3 | 0 | 2 | 5 | 4 |
| 19. | MC25019 | E–Commerce Technologies | 3 | 0 | 2 | 5 | 4 |
| 20. | MC25020 | Social Network Analysis | 3 | 0 | 0 | 3 | 3 |

| MC25101 | Mathematical Foundation for Computer Applications | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 1 | 0 | 4 |

**Course Objectives:**

- To apply fundamental concepts of discrete mathematics in the context of computer science and software development.
- To develop the ability to model and solve computational problems using mathematical tools such as combinatorics, matrices, graph theory, and number theory.
- To enhance analytical and logical thinking skills necessary for designing algorithms, analyzing complexity, and supporting various applications in computer science.

**Relations and Functions:** Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations – Inverse Function Composition of functions, recursive functions Lattices – Hasse diagram, Lattice and its Properties.

**Propositional and predicate Logic:** Propositional logic – Logical connectives – Truth tables – Normal forms (principal conjunctive and principal disjunctive normal forms) – Predicate logic – Universal and existential quantifiers – Proof techniques – Direct and indirect proofs – Proof by contradiction – Mathematical Induction.

**Recurrence Relation and Generating Function:** Generating Functions – Function of Sequences Calculating Coefficient of generating function, Recurrence relations – Solving recurrence relation by substitution and Generating funds. Characteristics roots solution of In homogeneous Recurrence Relation.

**Graph Theory:** Basic terminology: Different types of graphs – Directed and undirected, Simple, Pseudo, Complete, Regular, Bipartite. Incidence and degree, Pendant and Isolated vertex and Null graph. Isomorphism, Sub graphs, Walk, Path and Circuit, Connected and disconnected graphs and components, operations on graphs. Euler Graphs, Fleury's Algorithm – Hamiltonian circuits and paths. Traveling salesman problem. Matrix representation of graphs – Incidence and Adjacency matrices.

**Grammars and Automata:** Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA) – Formal Definition, Simplified notation: State transition graph, Transition table – Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition – Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata.

**Weightage:** Continuous Assessment: 40%, End Semester Examinations: 60%.

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

**References**:


1. Trembly, J. P., & Manohar, R. (2017). Discrete mathematical structures with applications to computer science (International ed.). Tata McGraw-Hill.
2. Rosen, K. H. (2017). Discrete mathematics and its applications (7th ed.). McGraw-Hill.
3. Deo, N. (2016). Graph theory with applications to engineering and computer science. Prentice Hall.
4. Hopcroft, J. E., & Ullman, J. D. (2006). Introduction to automata theory, languages, and computation (3rd ed.). Pearson Education.
5. Sharma, J. K. (2011). Discrete mathematics (3rd ed.). Macmillan Publishers India.


**E-resources:**

1. https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/?
2. https://teachyourselfcs.com/?

| MC25102 | Networking and Communication Systems | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objectives:**

- To provide a comprehensive understanding of networking principles and communication models of modern communication systems.
- To enable students to analyze, design, and evaluate computer networks and communication systems.
- To develop skills in applying networking and communication concepts to real-world applications

**Networking Fundamentals and Protocol Models:** Basics of storage, Need for modern storage, Overview of data growth and digital transformation, Features of DAS, NAS, SAN, CAS, Introduction to RAID: types, levels, and configurations, Disk subsystems, Performance and reliability considerations, Traditional vs. software-defined storage.

**Activity:** Create visual OSI vs TCP/IP Model Mapping Chart with real-world protocol examples at each layer.

**Data Link and Network Layer Protocols:** Data link layer services, design issues, error detection and correction, Medium Access Control (MAC): Ethernet, CSMA/CD, CSMA/CA, Network layer functions and IP routing, store and forward packet switching, connection less and connection, oriented networks, Routing algorithms: Distance vector, Link state, OSPF, BGP, ARP, ICMP, and DHCP operations

**Activity:** Simulate IP Routing in a Virtual LAN using Packet Tracer Tool.

**Transport and Application Layer Protocols:** Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP, Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol.

**Activity:** Demonstration of Transport and Application layer protocols.

**Wireless and Mobile Communication Systems:** Cellular networks and mobile IP,Wireless LANs and IEEE 802.11, Bluetooth, ZigBee, and NFC basics, Mobile communication challenges: handoff, roaming, latency, MANETs and VANETs

**Activity:** Realization of Blue Tooth and ZigBee Network

**Modern Network Trends, IoT, Cloud & Secure Networking:** Cloud networking and virtualization, IoT architecture and protocols (MQTT, CoAP), Edge and fog computing, Network security basics: firewalls, encryption, VPNs, Network challenges in modern computing (5G, SDN, NFV)

| |
|---|
| **Activity:** Demonstration of IoT Architecture for Home Automation |
| **Weightage:** Continuous Assessment: 40%, End Semester Examinations: 60%. |
| **Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20). |
| **References:**<br>1. Tanenbaum, A. S., & Wetherell, D. J. (2014). Computer Networks. Pearson Education, Elsevier Inc.<br>2. Gupta, P. C. (2014). Data communications and computer network. PHI Learning Pvt Ltd.<br>3. Forouzan, B. A. (2013). Introduction to data communication & networking. McGraw Hill Education Pvt Ltd.<br>4. Larry, P., Peterson, L., & Davis, B. S. (2014). Computer networks*:* A system approach. Elsevier Inc.<br>5. Forouzan, B. A. (2022). Data communications and networking. McGraw-Hill. |
| **E-RESOURCES:**<br>1. Computer Networks by Prof. S. K. Sinha, IIT Kharagpur https://nptel.ac.in/courses/106105183<br>2. Introduction to Wireless and Cellular Communications by IIT Madras https://nptel.ac.in/courses/117106114 |

| | **Description of CO** | **PO** | **PSO** |
|---|---|---|---|
| CO1 | Elaborate the fundamental concepts of data communication, OSI, and TCP/IP models. | -- | -- |
| CO2 | Apply transmission media, switching, and error control techniques to optimize communication efficiency | PO1(3) | PSO1(3) |
| CO3 | Design and simulate routing, congestion control, and network protocols for efficient communication systems. | PO3(2) | PSO2(2) |
| CO4 | Evaluate and justify emerging networking technologies and security mechanisms for real-world applications. | PO2(1) | PSO1(3) |

| MC25103 | Software Testing and Automation | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objectives:**

- To impart knowledge of software testing principles and life cycle models for ensuring software quality and reliability.
- To enable students to design effective test cases and utilize automation tools for functional and non-functional testing.
- To develop skills in implementing automated testing frameworks for real-world software projects.

**Foundations of Software Testing:** Foundations of Software Testing Introduction to test Software, Black-Box Testing and White-Box Testing-Software Testing Life Cycle, V- model of Software Testing-Program Correctness and Verification, Reliability versus Safety, Failures, Errors and Faults (Defects), Software Testing Principles, Program Inspections, Stages of Testing, Unit Testing-Integration Testing, System Testing.

**Activities:**

- Creation of Black-Box and White, Box test cases based on functional and Non-functional requirements.
- Simulation of Software Testing Life Cycle to understand the stages of testing

**Test Planning:** The Goal of Test Planning, High Level Expectations, Intergroup Responsibilities, Test Phases, Test Strategy, Resource Requirements, Tester Assignments, Test Schedule, Test Cases, Bug Reporting, Metrics and Statistics.

**Activities:**

- Design a comprehensive test plan for a sample software project,
- Bug Reporting and Metrics Simulation.

**Test Design and Execution:** Test Objective, Identification, Test Design, Factors, Requirement, identification, Testable, Requirements, Modeling a Test Design Process, Modeling Test Results, Boundary Value Testing, Equivalence Class Testing, Path Testing-Data Flow Testing, Test Design Preparedness Metrics, Test Case Design Effectiveness, Model, Driven Test Design, Test Procedures, Test Case Organization and Tracking-Bug Reporting, Bug Life Cycle.

**Activities:**

- Conduct load, stress, volume, fail-over, recovery, and configuration tests on a sample application
- Test a web or mobile application for usability, compatibility, security.

**Advanced Testing Concepts:** Performance Testing: Load Testing, Stress Testing, Volume Testing, Fail-Over Testing, Recovery Testing-Configuration Testing, Compatibility Testing, Usability Testing, Testing the Documentation, Security testing, Testing in the Agile Environment-Testing Web and Mobile Applications.

**Activities:**

- Perform configuration tests on a web or mobile application using performance testing tools
- Performing iterative testing in an Agile environment on web and mobile applications.

**Test Automation and Tools:** Automated Software Testing, Automate Testing of Web Applications-Selenium: Introducing Web Driver and Web Elements, Locating Web Elements, Actions on Web Elements, Different Web Drivers, Understanding Web Driver Events, Testing: Understanding Testing.xml, Adding Classes, Packages, Methods to Test-Test Reports.

**Activities:**

- Use Selenium WebDriver to generate test reports.
- Design automated test scripts on a sample web application

**Weightage:** Continuous Assessment: 40%, End Semester Examinations: 60%.

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

**References:**


1. Desikan, S., & Ramesh, G. “Software testing: Principles and practices”. Pearson Education.
2. Westerveld, D. (2021). “API testing and development with Postman”. Packt Publishing.
3. Gundecha, U. “Selenium Web Driver 3 practical guide”. Packt Publishing.
4. Rodrigues, A. G. “Mastering JMeter: From load testing to DevOps”. Packt Publishing.
5. Menon, V. “TestNG beginner’s guide”. Packt Publishing.


**E- Resources:**

1. NPTEL: Software Testing by Prof. Yogesh Singh
2. Postman Tutorials: Postman.com
3. JMeter Documentation: https://jmeter.apache.org/
4. TestNG Documentation: https://testng.org/doc/

| | Description of CO | PO | PSO |
|---|---|---|---|
| CO1 | Explain software testing principles and defect management to ensure software reliability. | -- | -- |
| CO2 | Apply test planning strategies using black-box and white-box techniques. | PO1(3) | PSO1(3) |
| CO3 | Develop automated test scripts to validate functionality, performance, and security. | PO3(2) | PSO2(2) |
| CO4 | Evaluate advanced testing methodologies to address real-world challenges. | PO2(1) | PSO1(3) |

| MC25104 | Data Structures and Algorithms using Python | L | T | P | C |
|---|---|---|---|---|---|
| | | 2 | 0 | 2 | 3 |

**Course Objectives:**

- To provide a strong understanding of fundamental data structures and their implementation in Python.
- To enable students to design, analyze, and implement efficient algorithms for searching, sorting, and problem-solving using Python.
- To develop skills in applying algorithmic thinking and data structure concepts to real-world computational problems

**Python Programming and Algorithmic Analysis:** Python environment, variables, operations, control flow, conditionals, loops, and functions, Python types and expressions, Strings, lists, and tuples, names, mutable and immutable values, List operations and slicing, Dictionaries, Functions, Object oriented programming, Abstract data types, Classes and objects, Encapsulation and data abstraction, Inheritance, Namespace and Object Orientation.

**Practical:**

- Student Record Management System (Procedural Approach)
- Bank Account Simulation Using OOP

**Abstract Data Types and Linear Structures:** Abstract Data Types (ADTs), ADTs and classes, introduction to OOP, classes in Python, inheritance namespaces, shallow and deep copying Introduction to analysis of algorithms, asymptotic notations divide & conquer, recursion, analyzing recursive algorithms- Stack ADT, Queue ADT

**Practical:**

- Implement Custom Stack and Queue ADTs with Recursion and Algorithm Analysis
- Recursive Merge Sort with Operation Count and Custom Object Sorting

**Sorting and Searching:** Sorting Algorithms: Bubble sort, selection sort, insertion sort, merge sort, quick sort, analysis of sorting algorithms, Searching Techniques: linear search, binary search, hashing, hash functions, collision handling, load factors, rehashing, and efficiency

**Practical:**

- Implement sorting algorithms in Python Bubble Sort**,** Selection Sort
- Implement Linear Search, Binary Search, Hash Table with Collision Handling

**Tree Structures:** Tree ADT, Binary Tree ADT, tree traversals,binary search trees, AVL trees, heaps, multiway search trees

**Practical:**

- lement Binary Tree ADT
- Implement an AVLTree class with Rotations (LL, RR, LR, RL)

**Graph Structures:** Graph ADT, representations of graph, graph traversals, DAG, topological ordering, greedy algorithms, dynamic programming, shortest paths, minimum spanning trees, Introduction to complexity classes and intractability

**Practical:**

- Implement Breadth-First Search (BFS), Depth-First Search (DFS)
- Implement Dijkstra's Algorithm for shortest path (Greedy)

**Weightage:** Continuous Assessment: 50%, End Semester Examinations: 50%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

**References:**


1. Agarwal, B., & Baka, B. (2018). Hands-on data structures and algorithms with Python: Write complex and powerful code using the latest features of Python 3.7. Packt Publishing.
2. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2021). Data structures & algorithms in Python (Indian adaptation). John Wiley & Sons.
3. Lee, K. D., & Hubbard, S. (2015). Data structures and algorithms with Python. Springer.
4. Necaise, R. D. (2011). Data structures and algorithms using Python. John Wiley & Sons.
5. Weiss, M. A. (2014). Data structures and algorithm analysis in C++. Pearson Education.


**E-resources:**

1. https://runestone.academy/ns/books/published/pythonds/index.html
2. https://www.geeksforgeeks.org/data-structures-and-algorithms-in-python/

| | Description of CO | PO | PSO |
|---|---|---|---|
| CO1 | Elaborate programming concepts and object-oriented paradigms to solve computational problems using python. | -- | -- |
| CO2 | Apply linear and nonlinear data structures to solve algorithmic problems efficiently. | PO1(3) | PSO1(3) |
| CO3 | Evaluate and implement classical algorithms for sorting, searching, traversing, and optimization with complexity analysis. | PO3(2) | PSO2(2) |
| CO4 | Design and implement Python-based applications using ADTs to solve real-world computational challenges | PO2(1) | PSO1(3) |

| MC25105 | Data Exploration and Visualization | L | T | P | C |
|---|---|---|---|---|---|
| | | 2 | 0 | 2 | 3 |

**Course Objectives:**

- To provide students with the ability to explore datasets to uncover patterns.
- To enable learners to create effective visualizations to communicate insights clearly.
- To develop skills in presenting data-driven insights for informed decision-making.

**Exploratory Data Analysis:** Identifying Data Quality, Missing Values, Irregular Cardinality, Outliers, Handling Data Quality, Describing Data, Preparing Data Tables, Understanding Relationships, Identifying and Understanding Groups, Building Models from Data, Significance, Classical and Bayesian Analysis

**Practical:**

- Identification of missing values and detection of outliers
- Creation of summary table and visualization of data distribution

**Univariate and Bivariate Analysis:** Distributions and Types of Variables, Numerical Summaries (Level and Spread), Scaling and Standardizing, Measures of Inequality, Smoothing Time Series, Relationships Between Two Variables, Percentage Tables, Analyzing Contingency Tables.

**Practical:**

- Generation of Plots and application of scaling.
- Creation of a scatterplot and interpret the relationship.

**Multivariate and Time Series Analysis:** Multivariate Analysis, Causal Explanations, Three, Variable Contingency Tables and Beyond, Longitudinal Data, Fundamentals of Time Series Analysis (TSA), Characteristics of Time Series Data, Data Cleaning, Time, Based Indexing, Grouping, Resampling.

**Practical:**

- Creation of a three–variable contingency table
- Time series data analysis for clean missing or inconsistent timestamps

**Data Visualization and Exploration:** Visualization Objectives and Key Factors, Data Representation and Presentation, Stages of Data Visualization, Visualization Tools and Widgets, Mapping and Geographic Data, Time Series Visualization, Correlations and Scatterplots, Trees, Hierarchies, Networks

**Practical:**

- Visualization of dataset using multiple subplot.
- Generation of correlation heatmap and a map-based plot u.

**Interactive Data Visualization:** Text and Document Visualization, Levels of Text Representations, Single Document Visualizations, Document Collection Visualizations, Interaction Concepts and Techniques, Designing Effective Visualizations, Comparing and Evaluating Visualization Techniques.

**Practical:**

- Visualization of word frequencies from a small text dataset.
- Creation of chart using interactive visualization libraries.

**Weightage:** Continuous Assessment: 50%, End Semester Examinations: 50%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

**References:**


1. Mukhiya, S. K., & Ahmed, U. (2020). Hands-on exploratory data analysis with Python. Packt Publishing.
2. Sharma, T. (2023). Mastering exploratory data analysis with Python: Gain a solid understanding of exploratory data analysis and data visualization techniques using Python. Packt Publishing.
3. Schwabish, J. (2021). Better data visualizations: A guide for scholars, researchers, and wonks. Columbia University Press.


**E-Resources:**

1. https://app.datacamp.com/learn/courses/statistical-thinking-in-python-part-1
2. https://realpython.com/python-data-visualization-bokeh

| | Description of CO | PO | PSO |
|---|---|---|---|
| CO1 | Explain the data analysis using statistical and exploratory techniques | -- | -- |
| CO2 | Apply univariate, bivariate, and multivariate analysis methods to extract insights from data. | PO1(3) | PSO1(3) |
| CO3 | Design effective visualizations to communicate complex data patterns. | PO3(2) | PSO2(2) |
| CO4 | Evaluate interactive visualizations for real-world applications | PO2(1) | PSO1(3) |

| MC25106 | Advanced Java Programming | L | T | P | C |
|---|---|---|---|---|---|
| | | 2 | 0 | 2 | 3 |

**Course Objectives:**

- To provide in-depth knowledge of advanced Java concepts.
- To enable students to develop Java applications using object-oriented and event-driven programming techniques.
- To develop proficiency in using Java frameworks, APIs, and libraries for building real-world applications.

**Core Java Fundamentals:** Java Virtual Machine, data types, variables, keywords, operators, expressions, control statements, classes, objects, constructors, access control, method overloading, static members, Arrays, Strings, Inheritance: types, constructors in inheritance, method overriding, use of super, abstract classes – interfaces, dynamic method dispatch. Packages and exception handling

**Practical:**

- Write a Java program that demonstrates the use of classes, objects, constructors, method overloading, inheritance, super, and abstract classes.
- Create a Java application that includes built–in and user–defined exception handling using try, catch, finally, throw, and throws.

**Multithreading, JDBC, and Web Client UI Development:** Java Thread Model, Concurrent Programming, Thread Life Cycle, Thread Priorities, Creating and Managing Threads, Thread Methods and Exceptions, Inter-thread Communication, Synchronization. JDBC Architecture and Drivers, CRUD Operations – Connecting to Relational Databases. Web Client UI: HTML and CSS, Responsive Web Design using Bootstrap, JavaScript, React JS, Creating Reusable UI Components with React.

**Practical:**

- Implement multiple threads in Java using the Thread class and Runnable interface. Demonstrate thread lifecycle, priorities, and synchronization.
- Connect to a relational database (MySQL/PostgreSQL) and perform Create, Read, Update, and Delete operations using JDBC

**Jave Servlets:** Java EE Architecture - Application Servers and Containers, Servlet API and Lifecycle, Handling Client Requests and Responses - Working with Databases through Servlets-Servlet Advanced Concepts: Request Dispatcher -Session Management, Cookies and Http Session, Working with Files, Non- blocking I/O in Servlets.

**Practical:**

- Implement a servlet for uploading files to the server and downloading them using Multipart Config and File Output Stream
- Develop a servlet to handle HTTP GET and POST methods. Accept form data (e.g., user registration) and display it.

**Java Server Pages and JSTL:** JSP Architecture and Lifecycle, Action Elements, Implicit Objects, Scripting Elements, Scope and EL (Expression Language), JSP Standard Tag Libraries (JSTL), Developing Dynamic Web Applications with JSP.

**Practical:**

- Design a responsive UI form using Bootstrap and React to capture user input and display the result
- Create a JSP application using scripting elements, EL for data display, and JSTL for conditional and loop–based rendering.

**Java Frame works:** MVC Architecture, Spring Framework, Spring Modules, Spring MVC and Spring Boot, Hibernate Framework, Other Popular Frameworks. Maven Installation, Maven Core Concepts - Basic Structure of the POM File, Maven Build Life Cycles, Phases and Goals - Maven Build Profiles

**Practical:**

- Build a Java application using Hibernate and JPA annotations to perform CRUD operations on a Student entity.
- Mini-project integrating Spring and Hibernate framework.

**Weightage:** Continuous Assessment: 50%, End Semester Examinations: 50%

**Assessment Methodology:** Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

**References:**


1. Schildt, H. (2019). Java: The complete reference (9th ed.). Tata McGraw Hill Publishing Company Limited.
2. Balagurusamy, E. (2017). Programming with Java: A primer (5th ed.). Tata McGraw Hill Publishing Company Limited.
3. Horstmann, C. S., & Cornell, G. (2013). Core Java volume I – Fundamentals (9th ed.). Pearson Education.
4. Schildt, H. (2014). Java: A beginner's guide (6th ed.). Tata McGraw Hill Publishing Company Limited.
5. Moraes, E. (2018). Java EE 8 cookbook: Build reliable applications with the most robust and mature technology for enterprise development. Packt Publishing.
6. Shah, S., & Shah, V. (2017). Java EE 7 for beginners. SPD.


**E-Resources:**

1. https://docs.oracle.com/javase/tutorial/
2. https://www.baeldung.com/java-thread
3. https://docs.oracle.com/javaee/7/tutorial/index.html

| | Description of CO | PO | PSO |
|---|---|---|---|
| CO1 | Elaborate advanced Java programming concepts to develop scalable software solutions. | -- | -- |
| CO2 | Design database-driven applications using JDBC, servlets, and JSP. | PO1(3) | PSO1(3) |
| CO3 | Develop dynamic web applications using Java frameworks and MVC architecture. | PO3(2) | PSO2(2) |
| CO4 | Integrate modern Java tools and frameworks to build secure enterprise-level applications. | PO2(1) | PSO1(3) |

| MC25201 | UI & UX Design | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objective:**

To provide a sound knowledge of UI and UX principles; to recognize the importance of user interface and user experience in product design; to explore various research methods applied in design thinking; to investigate and utilize tools commonly used in UI and UX processes; and to develop practical skills in creating wireframes and prototypes for user-centered design solutions.

**Essentials of Interaction Design**

UI vs. UX Design - Core Stages of Design Thinking - Divergent and Convergent Thinking - Brainstorming and Game storming - Observational Empathy

**Suggested Activities:**

**Activity 1:** Empathy Mapping Workshop
**Activity 2:** Game storming Session

**UI Design Fundamentals**

Visual and UI Principles - UI Elements and Patterns - Interaction Behaviors and Principles – Branding - Style Guides

**Suggested Activities:**

**Activity 3:** UI Component Audit
**Activity 4:** Create a Mini Style Guide

**User Research and Experience**

Introduction to User Experience - Why You Should Care about User Experience - Understanding User Experience - Defining the UX Design Process and its Methodology - Research in User Experience Design - Tools and Method used for Research - User Needs and its Goals - Know about Business Goals

**Suggested Activities:**

**Activity 5:** User Interview Simulation
**Activity 6:** Persona Creation

**Wireframing, Prototyping and Testing**

Sketching Principles - Sketching Red Routes - Responsive Design – Wireframing - Creating Wire flows - Building a Prototype - Building High-Fidelity Mockups - Designing Efficiently with Tools - Interaction Patterns - Conducting Usability Tests - Other Evaluative User Research Methods - Synthesizing Test Findings - Prototype Iteration

**Suggested Activities:**

**Activity 7:** Wireframe a Key Flow
**Activity 8:** Usability Testing Roleplay

**Research, Designing, Ideating, & Information Architecture**

Identifying and Writing Problem Statements - Identifying Appropriate Research Methods - Creating Personas - Solution Ideation - Creating User Stories - Creating Scenarios - Flow Diagrams - Flow Mapping - Information Architecture

**Suggested Activities:**

**Activity 9:** User Flow Mapping
**Activity 10:** Card Sorting for IA

**Course Outcomes:**

1. Analyze the core principles and differences between UI and UX design and apply design thinking approaches to user-centered product development.
2. Apply UI design principles such as visual hierarchy, consistency, branding, and interaction behavior in creating user interfaces.
3. Conduct effective user research using appropriate qualitative and quantitative methods to gather insights into user needs and business goals.
4. Design wireframes, prototypes, and mockups using industry-standard tools and evaluate them through usability testing.
5. Construct effective information architecture and user flows, and ideate design solutions based on user personas, problem statements, and research findings.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GA questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale, “Human-Computer Interaction”, 5th Edition, Pearson Education, 2023.
2. Rex Hartson and Pardha Pyla, “The UX Book: Agile UX Design for a Quality User Experience”, 2nd Edition, Morgan Kaufmann, 2019.
3. Joel Marsh,“UX for Beginners: A Crash Course in 100 Short Lessons”, O’Reilly Media, 2016.
4. Thomas Michaud, “Foundations of Web Design: Introduction to HTML & CSS”, New Riders, 2021.
5. Russ Unger and Carolyn Chandler, “A Project Guide to UX Design: For User Experience Designers in the Field or in the Making”, 2nd Edition, New Riders, 2019.


**E-RESOURCES:**

1. https://www.designkit.org
2. https://developer.apple.com/design/human-interface-guidelines/

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO 1** | Analyze the core principles and differences between UI and UX design and apply design thinking approaches to user-centered product development. | PO1(3), PO2(1), PO3(1), PO4(3), PO5(1) | 3 | 3 | 1 |
| **CO 2** | Apply UI design principles such as visual hierarchy, consistency, branding, and interaction behavior in creating user interfaces. | PO1(2), PO2(3), PO3(1), PO4(3), PO5(2) | 1 | 2 | 2 |
| **CO 3** | Conduct effective user research using appropriate qualitative and quantitative methods to gather insights into user needs and business goals. | PO1(1), PO2(3), PO3(3), PO4(2), PO5(2) | 1 | 3 | 3 |
| **CO 4** | Design wireframes, prototypes, and mockups using industry-standard tools and evaluate them through usability testing. | PO1(1), PO2(2), PO3(3), PO4(3), PO5(1) | 3 | 3 | 3 |
| **CO 5** | Construct effective information architecture and user flows, and ideate design solutions based on user personas, problem statements, and research findings. | PO1(1), PO2(2), PO3(3), PO4(2), PO5(1) | 3 | 2 | 2 |

| MC25202 | Foundations of Data Science | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Course Objectives:**

The objective of this course is to enable students to understand the fundamentals and processes of data science. It aims to develop the ability to describe and summarize data effectively within the data science workflow. Students will also learn to analyze relationships between different data variables. In addition, the course emphasizes the use of Python libraries for data wrangling and equips students with skills to present and interpret data using visualization tools in Python.

**Data Science Basics**

Data Science: Benefits and uses – facets of data - Data Science Process: Overview – Defining research goals – Retrieving data – Data preparation - Exploratory Data analysis – build the model– presenting findings and building applications - Data Mining - Data Warehousing – Basic Statistical descriptions of Data

**Suggested Activities:**

**Activity 1:** Case Study Discussion
**Activity 2:** Quiz / Crossword

**Practical Activities:**

1. Exploration of open data sets
2. Exploration of open-source analytics tool.

**Describing Data**

Types of Data - Types of Variables -Describing Data with Tables and Graphs –Describing Data with Averages - Describing Variability - Normal Distributions and Standard (z) Scores

**Suggested Activities:**

**Activity 3:** Graphical Analysis
**Activity 4:** Measure of Center Project
**Practical Activities:**

1. Study R Languages, Commands, etc Consider 80 observations(dataset), generating random data using functions provided, like rbinom, performing basic statistical computations using built-in functions of R.

2. Discussion of R graphics. Histograms. Stem and leaf plots. Boxplots. Scatterplots, Bar graphs plotting the data using line graph, histograms, multiple graphs, etc. Generate 3D graphs or plots.

3. Measures of Central Tendency: Given a sample of 50 Observations (from any dataset), use possible functions R or Python and calculate mean, sd, var, min, max, median, range, and quantile. Discuss the properties of this distribution. Generate bell curve of a random normal distribution.

**Describing Relationships**

Correlation –Scatter plots –correlation coefficient for quantitative data –computational formula for correlation coefficient – Regression –regression line –least squares regression line – Standard error of estimate – interpretation of r2 –multiple regression equations –regression towards the mean

**Suggested Activities:**

**Activity 5:** Correlation & Regression Lab
**Activity 6:** Interpretation Exercise

**Practical Activities:**

1. Reshaping a data frame. Basics of text processing. Reading unusual data files. Basics of variable coercion. Hypothesis testing and t-test for any given dataset. Find out null hypothesis, alternate hypothesis, draw the picture (graph) to visualize problem. Test the value of population mean.
2. State alpha level and rejection region, estimate the maximum likely hood and inference.

**Python Libraries for Data Wrangling**

Basics of NumPy arrays –aggregations –computations on arrays –comparisons, masks, Boolean logic – fancy indexing – structured arrays – Data manipulation with Pandas – data indexing and selection – operating on data – missing data – Hierarchical indexing – combining datasets – aggregation and grouping – pivot tables

**Suggested Activities:**

**Activity 7:** Data Cleaning Project
**Activity 8:** NumPy Logic Tasks

**Practical Activities:**

1. Binomial simulation: Making the computer flip coins for you, Make use of rbinom function of R to generate samples, and other functions: counts, avgs, mean, sd, sqrt, hist (histogram). Bayesian Hypothesis testing on any given dataset or dataframe.
2. Use seaborn and combines simple statistical fits with plotting on pandas dataframes. Working on Linear Algebra and Linear Systems

**Data Visualization**

Importing Matplotlib – Line plots – Scatter plots – visualizing errors – density and contour plots – Histograms – legends – colors – subplots – text and annotation – customization – three-dimensional plotting - Geographic Data with Basemap - Visualization with Seaborn.

**Suggested Activities:**

**Activity 9:** Matplotlib Hands-On
**Activity 10:** Seaborn Project

**Practical Activities:**

1. Working on Monte Carlo Integration (Quasi-random numbers and find out the variance on any dataframe)

**Course Outcome:**

1. Demonstrate the key components of the data science process, including data collection, preparation, analysis, and presentation.
2. Analyze and summarize various types of data using appropriate tables, graphs, and statistical descriptors.
3. Interpret relationships between variables using correlation and regression techniques, and evaluate model performance.
4. Apply Python libraries such as NumPy and Pandas for data wrangling, manipulation, and transformation.
5. Create effective and interactive visualizations using Matplotlib, Seaborn, and other visualization tools to communicate insights from data.

**Assessment weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>i. Activities: 15%<br>ii. Internal Theory Examinations: 30%<br>iii. Internal Laboratory Examinations: 15% | End Semester Theory Examination: 40% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Suresh Kumar Mukhiya, Usman Ahmed, “Hands–on Exploratory Data Analysis with Python”, Packt Publishing, 2020. ISBN: 9781838826034.
2. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, 2nd Edition, O'Reilly Media, 2022. ISBN: 9781098104030.
3. Joel Grus, “Data Science from Scratch: First Principles with Python”, 2nd Edition, O'Reilly Media, 2019. ISBN: 9781492041139.
4. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, 3rd Edition, O’Reilly Media, 2022. ISBN: 9781098104030.
5. Alberto Boschetti, Luca Massaron, “Python Data Science Essentials”, 3rd Edition, Packt Publishing, 2021. ISBN: 9781801072123.


**E-Resources:**

1. https://www.edx.org/course/data-8-foundations-of-data-science
2. https://pll.harvard.edu/course/data-science-r-basics

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Demonstrate the key components of the data science process, including data collection, preparation, analysis, and presentation. | PO1(3), PO2(2), PO3(2), PO4(1), PO5(3), PO6(1) | 3 | 2 | 2 |
| **CO2** | Analyze and summarize various types of data using appropriate tables, graphs, and statistical descriptors. | PO1(2), PO2(2), PO3(2), PO4(1), PO5(3), PO6(1) | 3 | 3 | 2 |
| **CO3** | Interpret relationships between variables using correlation and regression techniques, and evaluate model performance. | PO1(2), PO2(3), PO3(2), PO4(2), PO5(3), PO6(1) | 3 | 3 | 3 |
| **CO4** | Apply Python libraries such as NumPy and Pandas for data wrangling, manipulation, and transformation. | PO1(2), PO2(3), PO3(2), PO4(3), PO5(3), PO6(1) | 3 | 2 | 2 |
| **CO5** | Create effective and interactive visualizations using Matplotlib, Seaborn, and other visualization tools to communicate insights from data. | PO1(2), PO2(2), PO3(2), PO4(2), PO5(3), PO6(1) | 3 | 3 | 1 |

| MC25203 | Full Stack Web Development | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 4 | 5 |

**Course Objective:**

The course aims to provide students with a comprehensive understanding of core web technologies and modern frameworks essential for full stack development. It enables learners to design and develop responsive front-end applications using HTML, CSS, and JavaScript, and to implement robust backend APIs using Node.js and Express.js. Students will gain hands-on experience in managing data with MongoDB and integrating it effectively with backend logic. Additionally, the course covers deployment strategies for full-stack applications and emphasizes version control practices using Git and GitHub.

**Web Technologies and HTML Fundamentals**

Overview of web architecture – Frontend, Backend, Full Stack. Introduction to the MERN stack – MongoDB, Express.js, React, Node.js.Setting up the development environment – VS Code, npm, browser tools.HTML syntax, structure, tags, forms, media tags, meta tags, and accessibility basics.

**Suggested Activities:**

**Activity 1:** Flipped Classroom – Explore and present HTML elements and structure.

**Practical Activities:**

1. A platform where users can browse, register for, and manage events, while organizers can create and track event participation.
2. A personal finance tracker that allows users to log income and expenses, categorize transactions, and view analytical charts.

**CSS, Responsive Design, and Bootstrap**

CSS syntax, box model, selectors, typography, and layouts (flex, grid). Responsive design with media queries – mobile-first approach. Bootstrap framework – grid system, components (navbar, cards, buttons), and utility classes. Customizing Bootstrap themes and styling.

**Suggested Activities:**

**Activity 2:** Responsive webpage creation using Bootstrap grid system.

**Practical Activities:**

1. A learning management system where instructors can upload courses and students can enroll, access content, and take quizzes.
2. An appointment booking system for clinics or professionals, allowing users to schedule, confirm, and manage their appointments.

**JavaScript, ES6 Essentials & Version Control**

JavaScript variables, functions, arrays, objects, events, and DOM manipulation.ES6 features – arrow functions, template literals, destructuring, modules, and classes. Basic form validation and interactive UI elements. Version control – Git basics: init, clone, commit, push, pull, branch, merge.

**Suggested Activities:**

**Activity 3:** Hands-on Quiz – JavaScript DOM & Git operations.
**Practical Activities:**

1. A job portal where recruiters can post vacancies and applicants can search jobs, upload resumes, and apply online.
2. A complaint or support ticket system where users can raise issues, track their status, and receive responses from administrators.

**Node.js, Express.js and MongoDB**
Node.js fundamentals – npm, event-driven programming, creating a server.Express.js – routing, middleware, REST APIs, static file handling. Introduction to NoSQL with MongoDB – documents, collections, CRUD operations. Mongoose – schema design, validation, MongoDB-Node integration.

**Suggested Activities:**

**Activity 4**: Think-Pair-Share – Write and test CRUD APIs using Postman.
**Practical Activities:**

1. A library management application for searching, reserving, borrowing, and returning books, with inventory tracking for admins.

**React.js and Capstone Integration Project**
React fundamentals – JSX, components (functional/class), props, state, hooks (useState, useEffect). Routing with React Router – nested and dynamic routes. Form handling and validation in React. Integration – build a full-stack app (frontend + backend + database). Flutter basics – widgets, state management, and connecting to backend APIs. Overview of deployment platforms – Netlify, Render, Vercel, Heroku-React Native-flutter application development.

**Suggested Activities:**
**Activity 5:** Project-Based Learning – Build and present a mini MERN app.

**Practical Activities:**

1. A donation platform where users can create or contribute to fundraising campaigns, with live progress tracking and donor details.

**Course Outcomes:**

1. Analyze the architecture and core technologies used in full stack web development.
2. Design and develop responsive front-end interfaces using HTML, CSS, Bootstrap, and JavaScript.
3. Develop and integrate backend services and RESTful APIs using Node.js and Express.js.
4. Implement and manage database operations using MongoDB with Mongoose.
5. Deploy full stack applications and use Git/GitHub for source control and collaboration.

| **Assessment Weightage:** | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>(i). Activities: 15%<br>(ii). Internal Theory Examinations: 30%<br>(iii). Internal Laboratory Examinations: 15% | End Semester Theory Examination 40% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Jon Duckett, “HTML and CSS: Design and Build Websites”, Wiley, 2011.
2. Ethan Brown, “Web Development with Node and Express: Leveraging the JavaScript Stack”, O’Reilly Media, 2019 (2nd Edition).
3. Alex Banks and Eve Porcello, “Learning React: Functional Web Development with React and Redux”, O’Reilly Media, 2020 (2nd Edition).
4. Robin Nixon, “Learning Full-Stack JavaScript Development: MongoDB, Node.js, Express, and React”, McGraw-Hill Education, 2022


**E-RESOURCES:**

1. Mark Myers, “A Smarter Way to Learn JavaScript”, CreateSpace Independent Publishing Platform, 2016.
2. MDN Web Docs – https://developer.mozilla.org
3. React Official Documentation – https://reactjs.org/docs
4. MongoDB University Courses – https://university.mongodb.com
5. Node.js Documentation – https://nodejs.org/en/docs
6. Free Code Camp Full Stack Curriculum – https://www.freecodecamp.org

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the architecture and core technologies used in full stack web development. | PO1(3), PO2(2), PO3(2) | 3 | – | – |
| **CO2** | Design and develop responsive front-end interfaces using HTML, CSS, Bootstrap, and JavaScript. | PO1(3), PO2(2), PO3(3), PO5(2) | 3 | – | – |
| **CO3** | Develop and integrate backend services and RESTful APIs using Node.js and Express.js. | PO1(3), PO2(3), PO3(3), PO5(2) | 3 | – | – |
| **CO4** | Implement and manage database operations using MongoDB with Mongoose. | PO1(2), PO2(3), PO3(3) | 3 | – | – |

| | | | | | |
|---|---|---|---|---|---|
| **CO5** | Deploy full stack applications and use Git/GitHub for source control and collaboration. | PO1(2), PO3(2), PO5(3), PO6(2) | – | 3 | – |

| MC25204 | Mobile Application Development | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 4 | 5 |

**Course Objective:**

This course provides a hand–on introduction to building mobile applications, focusing on both native Android iOS platforms It covers platform-specific programming concepts, architecture, UI/UX design, and app deployment strategies. Through hands-on development using Android Studio and Xcode, students will gain experience in building native mobile apps that interact with databases, web services, and device features.

**Introduction to Mobile Platforms and Android Studio**

Android, iOS, and emerging platforms - Android platform architecture: Linux kernel, Libraries, Android Runtime, Application Framework, and Apps - Installing and configuring Android Studio and SDK tools - Creating the first Android project: app structure, AndroidManifest.xml, Gradle files - Android components: Activities, Services, Broadcast Receivers, Content Providers- Emulators vs real device deployment.

**Suggested Activities:**

**Activity 1:** Flipped Classroom, Quiz

**Practical Activities:**

1. Install and configure Java Development Kit (JDK), Android Studio, Android SDK, and Xcode. Create and deploy a basic 'Hello World' app on both Android and iOS simulators.

**UI/UX Design Principles and XML Layouts**

Layout managers: Linear Layout, Relative Layout, Constraint Layout, Frame Layout - UI widgets: Buttons, Text View, Edit Text, Image View, Check Box, Spinner - Event handling and input validation - Material Design principles and guidelines - Designing adaptive UI: screen densities, multi-language support, orientation handling - Introduction to themes, styles, drawable resources, and custom views.

**Suggested Activities:**

**Activity 2:** Flipped Classroom, Quiz

**Practical Activities:**

1. Design a user interface using basic widgets: buttons, text fields, image views, and spinners in Android. Create an equivalent UI layout in iOS using Storyboard and UIKit.
2. Develop a mobile app that responds to user interactions using event listeners. Demonstrate layout switching, form submission, and feedback using Android's XML layouts and iOS View Controllers.

**Activity Lifecycle, Intents, and Navigation**

Activity lifecycle: onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy() - Explicit and Implicit Intents: launching apps and components - Passing data between activities using Bundle, Serializable, Parcelable - Using Fragments for reusable UI components - Introduction to Jetpack Navigation Component for seamless screen transitions - Back-stack management and deep linking.

**Suggested Activities:**

**Activity 3:** Seminar

**Practical Activities:**

1. Implement a four-function calculator that performs addition, subtraction, multiplication, and division. Use platform-specific UI controls, and validate user input in both Android and iOS versions.
2. Create custom views and animations using Android's Canvas and Animator classes. Implement transitions and interactive animations in iOS using Swift and UIView animation APIs.

**Data Persistence and SQLite**

Shared Preferences: storing simple key-value pairs - Working with SQLite databases using SQLite Open Helper - Introduction to Room Database: Entities, DAOs, and Live Data - Performing CRUD operations on local databases - Handling migrations and database versioning- Best practices for secure and efficient data storage.

**Suggested Activities:**

**Activity 4:** Project-Based

**Practical Activities:**

1. Create an application to perform insert, update, delete, and display operations using SQLite in Android. Use Core Data or SQLite in iOS for basic CRUD operations with a local database.
2. Build an app that integrates Firebase to send SMS or email notifications from an Android device. Use Firebase Messaging or Apple Push Notification Service (APNs) for sending alerts in iOS.

**Web API Integration and Networking**

Basics of HTTP/HTTPS protocols and API formats (JSON/XML) - Consuming RESTful APIs using Retrofit and Volley - JSON parsing and serialization/deserialization - Error handling, timeouts, and retry mechanisms - API authentication (API keys, OAuth basics - Integrating third-party APIs (e.g., Google Maps, Firebase)

**Suggested Activities:**

**Activity 5:** Reproduction of Research Paper

**Practical Activities:**

1. Implement file reading/writing on external storage (SD card) in Android. Use Notification Manager to display persistent and time-based notifications. Implement similar functionality using iOS File Manager and Notification Center.

**iOS Application Development**

Introduction to iOS architecture and MVC pattern - Xcode interface, project structure, and build settings - Swift programming basics: data types, control flow, optionals, functions, closures - Storyboards and Interface Builder: designing UI for iPhone/iPad – View Controllers, Segues, and Navigation Controller for screen transitions - Connecting UI to

code with IBOutlets and IBActions- Debugging, testing, and running on simulator or real device.

**Suggested Activities:**

**Activity 6:** Poster presentation

**Practical Activities:**

1. Create a location-aware Android app using Google Maps SDK that displays user position and geofencing alerts. Develop a similar iOS app using Core Location framework to track GPS, region monitoring, and motion detection.
2. Design a simple 2D game (e.g., tapping or matching game) using Android's Canvas and View logic. Create an equivalent mini game or interactive utility in iOS using SpriteKit or UIKit Dynamics.

**Course Outcomes:**

1. Construct the architecture and development environments of Android and iOS platforms.
2. Design responsive and user-friendly interfaces using XML (Android) and Storyboards (iOS).
3. Implement functional mobile applications that manage activity/view lifecycle and handle navigation.
4. Integrate local databases and web APIs into mobile apps using platform-specific libraries.
5. Build, test, and deploy native mobile applications using Android Studio and Xcode.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>(i). Activities: 15%<br>(ii). Internal Theory Examinations: 30%<br>(iii). Internal Laboratory Examinations: 15% | End Semester Theory Examination: 40% |

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Barry Burd, Android Application Development for Dummies, 5th Edition, Wiley, 2023.
2. Joseph Annuzzi Jr., Lauren Darcey, Shane Conder, Android Programming: The Big Nerd Ranch Guide, 4th Edition, Addison-Wesley, 2022.
3. John Horton, Android Programming for Beginners, 3rd Edition, Packt Publishing, 2023.
4. Matt Neuburg, Programming iOS 14 with SwiftUI, O'Reilly Media, 2021.
5. Raywenderlich Tutorial Team, iOS Apprentice (Swift 5.5), Razeware LLC, 2022.

**E-Resources:**

1. Android Developers: https://developer.android.com
2. NPTEL Android Course: https://archive.nptel.ac.in/courses/106/106/106106156/
3. Kotlin Docs: https://kotlinlang.org/docs/home.html

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the architecture and development environments of Android and iOS platforms. | PO1(2), PO2(1), PO3(3), PO4(2), PO5(1), PO6(1) | 3 | 2 | 1 |
| **CO2** | Design responsive and user-friendly interfaces using XML (Android) and Storyboards (iOS). | PO1(2), PO2(1), PO3(3), PO4(3), PO5(1), PO6(2) | 3 | 3 | 2 |
| **CO3** | Implement functional mobile applications that manage activity/view lifecycle and handle navigation. | PO1(3), PO2(1), PO3(3), PO4(3), PO5(2), PO6(1) | 3 | 3 | 2 |
| **CO4** | Integrate local databases and web APIs into mobile apps using platform-specific libraries. | PO1(2), PO2(1), PO3(3), PO4(3), PO5(2), PO6(2) | 3 | 2 | 3 |
| **CO5** | Build, test, and deploy native mobile applications using Android Studio and Xcode. | PO1(2), PO2(3), PO4(3), PO5(1), PO6(2) | 2 | 2 | 3 |

| MC25205 | Quantum Computing | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisite:** Programming Data Structures and Algorithm using Python, Full Stack Web Development

**Course Objective:**
The objective of this course is to provide the basics of Robotic Process Automation and its role in digital transformation and various automation activities using RPA tools such as UiPath.

**Robotic Process Automation and UiPath**
Overview: Scope and Techniques of RPA – Benefits of RPA – Components of RPA – RPA Platforms – UiPath – UiPath Stack – Sequencing the Workflow – Activities – Control Flow, various types of loops and decision making.

**Suggested Activities:**

**Activity 1:** Quiz / Crossword on RPA Components and Platforms
**Activity 2**: Demonstrate decision–making and looping structures in UiPath

**UI Automation**
UI Automation: Activities Reference – Use Application/Browser – Go to URL – NavigatE-Browser – Highlight– Take Screenshot – CheckAppState – Click – TypeInto – Select Item – Check/Uncheck – Get Text – Get Attribute – Extract Table Data – Hover – Keyboard Shortcuts – Get Active Window – Maximize Window – Minimize Window – Hide Window – Restore Window – Move Window – App/Web Recorder.

**Suggested Activities:**

**Activity 3**: Automate filling out and submitting an online form.
**Activity 4**: Seminar on UI Events

**Mail Automation and Word Automation**
Mail Automation: Activities Reference – Desktop Outlook Setup – File System Structure – Use Gmail – For Each Email – Mark Email as Read/Unread – Forward Email – Save Email Attachments – Save Email – Send Email – Send Calendar Invite – Move Email – Reply to Email – Archive Email – Delete Email. Word Automation: Word Setup – File System Structure – Activities Reference: Use Word File – Save Document As – Read Text –Set Bookmark Content – Replace Text in Document – Append Text – Insert Data Table in Document – Replace Picture – Add Picture – Save Document as PDF.

**Suggested Activities:**

**Activity 5**: Generate a report with dynamic data in tabular format.
**Activity 6**: Presentation on Mail Automation activities.

**Excel Automation**
Excel Automation: Activities Reference – Use Excel File – Insert Sheet – Rename Sheet – Duplicate Sheet – Delete Sheet – For Each Excel Sheet – Insert Column – Text To Columns – Delete Column – Insert Rows – Delete Rows – Find 1st/Last Data Row – For Each Excel Row – Write Cell – Create Pivot Table – Format as Table – Change Pivot

Data Source – Refresh Pivot Table – Append Range – Copy Range – Sort Range – Clear Sheet/Range/Table, Auto Fill – Fill Range – Write Range – Read Cell Formula – Read Cell Value – Format Cells – Export to CSV – Save Excel File – Save Excel File As – Save Excel File As PDF.

**Suggested Activities:**
**Activity 7:** Create a workbook and manage multiple sheets automatically.
**Activity 8:** Seminar on activities of Excel sheet automation.

**CSV Automation and File Automation**
CSV Automation: Write CSV – Append to CSV – Read CSV. File Automation: Activities Reference: Get Folder Info – Folder Exists – Create Folder – Delete Folder – Copy Folder – Move Folder – For Each File in Folder – Compress/Zip Files – Extract/UnzipFiles – GetFileInfo – FileExists – CreateFile – DeleteFile – Copy File – Move File – Write Text File – Append Line – Read Text File.

**Suggested Activities:**
**Activity 9**: Read a CSV file and display the records based on a specific condition.
**Activity 10:** Group Discussion on the working of UiPath for File Automation.

**Course Outcomes:**

1. Analyze the key concepts, components, and benefits of Robotic Process Automation and UiPath architecture.
2. Design and develop workflows using UiPath to automate repetitive UI-based tasks with decision-making and control structures.
3. Implement automation solutions for mail and word-processing activities using predefined UiPath activities.
4. Develop automated Excel-based solutions, including pivot tables, formatting, data manipulation, and reporting tasks.
5. Construct and manage file system automation tasks using UiPath for CSV, text, and folder/file operations.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Tom Taulli, “The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems”, 2020, ISBN-13 (electronic): 978-1-4842-5729-6, Apress.
2. Alok Mani Tripathi, “Learning Robotic Process Automation: Create Software Robots and Automate Business Processes with the Leading RPA Tool – UiPath”, 2nd Edition, 2020, ISBN-13: 978-1838643745, Packt Publishing.
3. Rajeev Heda, “Robotic Process Automation Using UiPath StudioX: A Citizen Developer’s Guide to Hyperautomation”, 2022, ISBN-13: 978-1484286407, Apress.

4. Arun Kumar Asokan, “Mastering UiPath: An Advanced Guide to Robotic Process Automation”, 2021, ISBN-13: 978-9390684942, BPB Publications.
5. Srikanth Merianda, “Robotic Process Automation with UiPath: A Comprehensive Guide”, 2021, ISBN-13: 979-8704438284, Independently Published.

**E- RESOURCES:**

1. https://docs.uipath.com/activities/docs/ui-automation-activities
2. https://docs.uipath.com/activities/docs/csv-automation

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the key concepts, components, and benefits of Robotic Process Automation and UiPath architecture. | PO1(3), PO2(2), PO12(2) | 2 | 2 | 1 |
| **CO2** | Design and develop workflows using UiPath to automate repetitive UI-based tasks with decision-making and control structures. | PO1(3), PO2(3), PO3(2), PO5(2) | 3 | 2 | 2 |
| **CO3** | Implement automation solutions for mail and word-processing activities using predefined UiPath activities. | PO1(2), PO2(3), PO5(2), PO10(1) | 2 | 3 | 2 |
| **CO4** | Develop automated Excel-based solutions, including pivot tables, formatting, data manipulation, and reporting tasks. | PO2(3), PO3(2), PO5(3), PO10(2) | 3 | 3 | 2 |
| **CO5** | Construct and manage file system automation tasks using UiPath for CSV, text, and folder/file operations. | PO1(2), PO2(2), PO5(3), PO12(1) | 2 | 3 | 2 |

| MC25301 | Machine Learning Techniques | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Course Objective:**

To provide students with a comprehensive understanding of machine learning concepts, from basic principles to advanced techniques, enabling them to build, optimize, and evaluate ML models. The course emphasizes handson learning using modern tools, promotes independent exploration of ML research, and encourages ethical deployment of AI technologies.

**Introduction to Machine Learning**

Definitions and types of ML: Supervised- Unsupervised- Reinforcement Learning. Machine Learning workflow: Data preprocessing- feature engineering- model selection. Bias–Variance tradeoff. Tools and libraries: Python-scikit–learn- pandas-matplotlib-Jupyter/Colab.

**Suggested Activities:**

**Activity 1**: Flipped Classroom on ML lifecycle and tool introduction.

**Practical Activities:**

1. Explore the ML Workflow: Perform data preprocessing, feature engineering, and visualization using Python libraries (pandas, matplotlib, scikit-learn).
2. Implement Supervised Learning Models: Apply Linear Regression and Logistic Regression on real-world datasets.

**Supervised Learning Techniques**

Linear and Logistic Regression – Decision Trees and Random Forests- Naïve Bayes Classifier-k-Nearest Neighbors- Support Vector Machines (SVM).Evaluation metrics: Confusion Matrix- Accuracy- Precision- Recall- F1-Score. ROC-AUC – Validation techniques: Train-test split- k-Fold cross-validation.

**Suggested Activities:**

**Activity 2:** Quiz and case-based discussion on algorithm selection and evaluation.

**Practical Activities:**

1. Build and Evaluate Classifiers: Use Naïve Bayes, k-Nearest Neighbors (k-NN), and Support Vector Machines (SVM) with performance metrics (confusion matrix, precision, recall, F1-score).

**Unsupervised Learning & Dimensionality Reduction**

Clustering: k–Means- Hierarchical.DBSCAN – Dimensionality Reduction- PCA- Feature Selection and Transformation- Instance-Based Learning-t-SNE: Nonlinear visualization for high-dimensional data- Use in data exploration. Applications in real–world domains.

**Suggested Activities:**

**Activity 3:** Poster presentation or seminar on real–world use cases.

**Practical Activities:**

1. Unsupervised Learning Techniques: Apply k-Means, Hierarchical Clustering, on unlabelled datasets; visualize and compare clusters.
2. Dimensionality Reduction and Visualization: Perform PCA and t-SNE for reducing and visualizing high-dimensional data.

**Neural Networks- Optimization & Ethics**

Perceptron and activation functions- Error Propagation-BPN. Introduction to Deep Learning: ANN and CNN – Model optimization: L1/L2 regularization-dropout-Gradient Descent Optimization- Grid Search- Random Search – Interpretability: Introduction to SHAP- LIME – Ethical AI: Bias- fairness- responsible AI practices.

**Suggested Activities:**

**Activity 4:** Model tuning assignment and AI ethics case study discussion.

**Practical Activities:**

1. Neural Network Implementation: Build a simple Artificial Neural Network (ANN) using Tensor Flow/Keras for image or tabular classification tasks.
2. Model Optimization and Explainability: Use L1/L2 regularization and hyper parameter tuning (Grid Search/Random Search); interpret model results using SHAP or LIME

**Machine Learning Applications & Emerging Trends**

ML Applications: NLP- Healthcare- Finance- Image Recognition – Research paper review and critical analysis – Introduction to Reinforcement Learning and its use cases – Overview of Explainable AI-AutoML- and Federated Learning-GPT.

**Suggested Activities:**

**Activity 5:** “ML Think Tank” – Brainstorming session on innovative ML solutions.

**Practical Activities:**

1. Predict the likelihood of diabetes **or** heart disease using logistic regression or decision trees.
2. Build a CNN model to classify handwritten digits (MNIST dataset)

**Course Outcomes:**

1. Investigate real–world ML applications and assess ethical concerns in model deployment.
2. Design, train, and optimize neural networks using modern ML frameworks.
3. Apply unsupervised learning techniques for clustering and dimensionality reduction.
4. Build and evaluate supervised learning models using various performance metrics.
5. Demonstrate understanding of ML workflows, tools, and best practices

<table>
<tr><td colspan="3">Assessment Weightage:</td></tr>
<tr><td rowspan="2">Weightage :</td><td>Continuous Assessment: 60%</td><td rowspan="2">End Semester Theory Examination: 40%</td></tr>
<tr><td>(i).Activities: 15%<br>(ii).Internal Theory Examinations: 30%<br>iii).Internal Laboratory Examinations: 15%</td></tr>
<tr><td colspan="3">Mandated Activities with marks:<br><br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).</td></tr>
<tr><td colspan="3">Internal Examinations: TWO tests</td></tr>
</table>

**REFERENCES:**


1. AurélienGéron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 3rd Edition, O'Reilly Media, 2022.
2. Tom M. Mitchell, “Machine Learning”, 1st Edition, McGraw-Hill Education, 1997.
3. EthemAlpaydin, “Introduction to Machine Learning”, 4th Edition, MIT Press, 2020.
4. Ian Goodfellow, YoshuaBengio, and Aaron Courville, “Deep Learning”, MIT Press, 2016
5. Cathy O’Neil and Rachel Schutt, “Doing Data Science”, O'Reilly Media, 2013.


**E-RESOURCES:**

1. NPTEL: Machine Learning by Prof.SudarshanIyengar (IIT Ropar)
2. Google Machine Learning Crash Course
3. Scikit–learn Documentation
4. Keras Documentation
5. Edutech:https://www.edutechlearning.com/learningresoursedesc/computer-science/cse-lab-solutions/machine-learning-lab-resource

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Investigate real–world ML applications and assess ethical concerns in model deployment. | PO1(2), PO3(3), PO4(2) | 3 | 2 | 1 |
| **CO2** | Design, train, and optimize neural networks using modern ML frameworks. | PO1(2), PO3(3), PO4(3), PO6(2) | 3 | 3 | 2 |
| **CO3** | Apply unsupervised learning techniques for clustering and dimensionality reduction. | PO1(3), PO3(3), PO4(3), PO5(2) | 3 | 3 | 2 |
| **CO4** | Build and evaluate supervised learning models using various performance metrics. | PO3(3), PO4(3), PO5(2), PO6(2) | 3 | 2 | 3 |
| **CO5** | Demonstrate understanding of ML workflows, tools, and best practices. | PO2(3), PO4(3), PO6(2) | 2 | 2 | 3 |

| MC25001 | Big Data Analytics | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites:** SQL/Database Knowledge, Java/Scala/Python Programming

**Course Objective:**
To understand the need of Big Data, challenges and different analytical architectures, installation and understanding of Hadoop and its ecosystems. The Processing of Big Data with Advanced architectures like Hive, Pig and statistical R Programming.

**Introduction to Big Data and Hadoop**
Types of Digital Data – Introduction to Big Data – Challenges of conventional systems – Web data – Evolution of Analytic scalability – Analytic Processes and Tools – Analysis vs Reporting – History of Hadoop – Apache Hadoop – Analyzing Data with Hadoop.

**Suggested Activities:**

**Activity 1:** Assignment on Setting up and Installing Hadoop

**HDFS & Hadoop I/O**
Hadoop Distributed File System: The Design of HDFS– HDFS Concepts – The Command – Line Interface – Hadoop File Systems – Data Flow – Parallel Copying with DistCp – Hadoop Archives – Hadoop I/O: Data Integrity – Compression – Serialization.

**Suggested Activities:**

**Activity 2:** Seminar on HDFS Command Reference
**Activity 3:** Develop a Project on Listing contents of directory, Displaying and printing disk usage, Moving files & directories, Copying files and directories
**Activity 4:** Develop a Project to implement the file management tasks in Hadoop: Writing a file into HDFS, Reading data from HDFS, Retrieving files, Deleting files

**Hadoop and MapReducer**
Analyzing the Data with Hadoop- MapReduce Types- Input Formats- Output Formats –MapReduce Features - MapReduce Works - Anatomy of a MapReduce Job Run – Failures – Job Scheduling – Shuffle and Sort – Task Execution.

**Suggested Activities:**

**Activity 5:** Develop a Project on basic Word Count Map Reduce program to understand Map Reduce Paradigm with Weather Dataset

**Hadoop Related Tools**
Pig: Introduction to PIG- Execution Modes of Pig-Comparison of Pig with Databases-Grunt-Pig Latin- User Defined Functions, Data Processing operators-Hive: Hive Shell, Hive Services-Hive Metastore-Comparison with Traditional Databases-HiveQL- Tables-Querying Data-Data Ingestion and Management tools-Sqoop-Flume-Zookeeper-OoZie-Mahout.

**Suggested Activities:**

**Activity 6:** Install and Run Pig then write Pig Latin scripts to sort, group, join your data
**Activity 7:** Install and Run Hive then use Hive to create, alter, and drop databases, tables

**R Programming**

History and overview of R – Overview, Programming structures: Data Types – Control statements – Operators – Functions – Recursion – Replacement functions, R data structures – Vectors –Matrices and arrays – Lists –Data frames – Classes, Input/output, String manipulations.

**R Programming**

History and overview of R – Overview, Programming structures: Data Types – Control statements – Operators – Functions – Recursion – Replacement functions, R data structures – Vectors –Matrices and arrays – Lists –Data frames – Classes, Input/output, String manipulations.

**Suggested Activities:**

**Activity 8:** Learn the basics of functions in R and implement them with examples
**Activity 9: Implement** different data structures in R
**Activity10:** Implement different String Manipulation functions in R

**Course Outcomes:**

1. Analyze the characteristics of Big Data and the architecture of Hadoop ecosystems for processing large-scale datasets.
2. Execute HDFS commands and perform file operations such as read, write, copy, and delete within a Hadoop environment.
3. Design and implement MapReduce programs to solve real-world data analysis problems using distributed processing.
4. Develop data querying scripts using Pig Latin and HiveQL to manipulate and analyze structured data.
5. Construct data analysis workflows in R programming using functions, data structures, and string manipulations for statistical computing

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. V. K. Jain, Big Data and Hadoop, Khanna Publishing House, 2022.
2. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly Media, 2020.
3. Alan Gates, Programming Pig, O'Reilly Media, 2nd Edition, 2021.
4. Edward Capriolo, Dean Wampler, and Jason Rutherglen, Programming Hive, O'Reilly Media, 2020.
5. Garrett Grolemund, Hands-On Programming with R: Write Your Own Functions and Simulations, O'Reilly Media, 2021.


**E-RESOURCES:**

1. https://nptel.ac.in/courses/106/104/106104189

**CO PO Mapping**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the characteristics of Big Data and the architecture of Hadoop ecosystems for processing large-scale datasets. | PO1(3), PO3(2), PO4(2) | 3 | 3 | - |
| **CO2** | Execute HDFS commands and perform file operations such as read, write, copy, and delete within a Hadoop environment. | PO1(3), PO3(2), PO4(2) | 3 | 2 | - |
| **CO3** | Design and implement MapReduce programs to solve real-world data analysis problems using distributed processing. | PO1(3), PO3(3), PO4(2) | 3 | 3 | - |
| **CO4** | Develop data querying scripts using Pig Latin and HiveQL to manipulate and analyze structured data. | PO1(3), PO3(3), PO4(2) | 3 | 3 | - |
| **CO5** | Construct data analysis workflows in R programming using functions, data structures, and string manipulations for statistical computing | PO1(3), PO3(2), PO4(2), PO5(1) | 3 | 3 | - |

| MC25002 | Vibe Coding | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites -** pp Data Structures and Algorithms using Python

**Course Objective:**

The course aims to enhance creative and expressive programming through music, art, and real-time interactivity; introduce frameworks and tools used in live coding environments; develop coding skills that integrate sound synthesis, visual rendering, and sensory feedback; apply vibe coding techniques for real-time collaborative, artistic, and biometric applications; and explore security and privacy considerations in interactive coding environments used in performance contexts.

**Introduction To Vibe Coding**

Definition – Origins of Live Coding – Real-Time Creative Expression – Applications in Music, Visual Arts, Biometric Systems – Tools: Sonic Pi, TidalCycles, Hydra – IDEs for creative coding.

**Suggested Activities:**

**Activity 1:** Hands-on Demo – Create a basic musical loop using Sonic Pi or TidalCycles and demonstrate live changes.
**Activity 2:** Group Discussion – Explore the evolution of live coding from traditional programming to real-time artistic expression.

**Coding For Audio-Visual Systems**

Sound Synthesis – Audio Loops – Visual Output with Hydra – Shader Programming – Generative Art – Signal Processing for Real-Time Interaction – Synchronization of Visual and Audio Streams.

**Suggested Activities:**

**Activity 3:** Mini Project – Use Hydra to create a generative visual synchronized to an audio beat.
**Activity 4:** Shader Lab – Modify a basic GLSL shader to respond to sound amplitude in real-time.

**Sensory Integration And Biometrics**

Integration of biometric sensors (EEG, EMG, heart rate) – Data mapping for expressive coding – Haptic feedback – Emotion-based coding – Case studies on therapeutic and surveillance applications.

**Suggested Activities:**

**Activity 5:** Sensor Demo – Integrate a simple biometric sensor (heart rate or EEG) with a creative coding environment to change visuals based on signal input.
**Activity 6:** Case Study Presentation – Present real-world applications of biometric-integrated vibe coding in therapy or surveillance.

**Security And Collaborative Coding**

Multi-user Vibe Coding Systems – Secure Session Sharing – Real-Time Encryption of Code Streams – Identity Authentication in Live Performance – Versioning and Auditability – Cybersecurity in generative environments.

**Suggested Activities:**

**Activity 7:** Simulation – Use a collaborative coding tool (like Estuary or Hydra multi-user mode) and simulate a secure coding session.

**Activity 8:** Debate – "Should live code performances be encrypted?" Explore the balance between openness and security.

**Frameworks, Ethics And Future Trends**

Platforms: Sonic Pi, FoxDot, Max/MSP, OpenFrameworks – Privacy in Creative Code – AI-assisted vibe coding – Ethical Concerns in Manipulative Code Art – Future of human-centered programmatic expression.

**Suggested Activities:**

**Activity 9:** Tool Review – Compare Sonic Pi, FoxDot, and Max/MSP in terms of features, ease of use, and performance contexts

**Activity 10:** Poster Presentation – Ethical implications of AI-assisted generative art: privacy, authorship, and bias.

**Course Outcomes:**

1. Demonstrate the ability to write real-time code for audio-visual expression using live coding platforms such as Sonic Pi and Hydra.
2. Develop interactive systems by integrating biometric sensors and sensory feedback for immersive coding experiences.
3. Construct synchronized sound and visual outputs using generative and shader programming techniques.
4. Collaborate in multi-user live coding environments with security, encryption, and identity management features.
5. Analyze ethical, privacy, and creative implications in the use of vibe coding frameworks for performance and real-world applications.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15% | |
| | Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Thor Magnusson, "Sonic Writing: Technologies of Material, Symbolic and Signal Inscriptions", Bloomsbury Academic, 2019.
2. Alex McLean, Geraint Wiggins, "The Oxford Handbook of Algorithmic Music", Oxford University Press, 2021.
3. Eli Field steel, "Creative Live Coding with Supercollider", Routledge, 2023.
4. Nick Collins, Julio d'Escriván, "The Cambridge Companion to Electronic Music", 2nd Edition, Cambridge University Press, 2021.
5. Ge Wang, "Artful Design: Technology in Search of the Sublime", Stanford University Press, 2018.


**E-RESOURCES:**

1. https://thebookofshaders.com

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Demonstrate the ability to write real-time code for audio-visual expression using live coding platforms such as Sonic Pi and Hydra | PO1(2), PO3(2), PO6(2) | 2 | 3 | - |
| **CO2** | Develop interactive systems by integrating biometric sensors and sensory feedback for immersive coding experiences | PO1(3), PO3(3), PO4(2) | 3 | 3 | - |
| **CO3** | Construct synchronized sound and visual outputs using generative and shader programming techniques. | PO1(3), PO3(2), PO4(3), PO5(2) | 3 | 3 | - |
| **CO4** | Collaborate in multi-user live coding environments with security, encryption, and identity management features. | PO2(2), PO4(3), PO5(2), PO6(3) | 2 | 2 | 2 |
| **CO5** | Analyze ethical, privacy, and creative implications in the use of vibe coding frameworks for performance and real-world applications. | PO2(2), PO4(2), PO5(2), PO6(2) | 2 | 3 | 2 |

| MC25003 | Network Programming and Management | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites - Networking and Communication Systems**

**Course Objective:**

The course aims to provide an in–depth understanding of network protocols, APIs, and tools for network application development and management. Students will learn to create networked applications using sockets, perform traffic analysis, and understand network security principles. Emphasis is given to real-world protocols and cloud/network monitoring tools.

**Sockets and Application Development**

Introduction to Socket Programming – System Calls – Address Conversion Functions – POSIX Signal Handling – Server with Multiple Clients – Boundary Conditions – Server Process Crashes, Server Host Crashes, Server Crashes and Reboots, Server Shutdown – I/O Multiplexing – I/O Models – TCP Echo Client/Server with I/O Multiplexing.

**Suggested Activities:**

**Activity 1:** Develop a simple TCP client-server application where the server handles multiple clients using I/O multiplexing (select() or poll() system call).

**Activity 2:** Simulate and analyze the behavior of a server under different failure scenarios (e.g., host crash, server process crash, unexpected shutdown) and demonstrate graceful recovery.

**Socket Options**

Socket Options – getsockopt and setsockopt Functions – Generic Socket Options – IP Socket Options – ICMP Socket Options – TCP Socket Options – Multiplexing TCP and UDP Sockets – Domain Name System – gethostbyname, gethostbyaddr, getservbyname and getservbyport functions – Protocol Independent Functions – getaddrinfo and freeaddrinfo Functions.

**Suggested Activities:**

**Activity 3:** Create a C program to modify socket behavior using setsockopt(), such as enabling SO_REUSEADDR, setting TCP_NODELAY, and changing buffer sizes.

**Activity 4:** Write a program that performs domain name resolution using gethostbyname(), getaddrinfo(), and demonstrate interoperability with IPv6.

**Advanced Sockets**

IPv4 and IPv6 Interoperability – Threaded Servers – Thread Creation and Termination – TCP Echo Server using Threads – Mutex – Condition Variables – Raw Sockets – Raw Socket Creation – Raw Socket Output – Raw Socket Input – Ping Program – Trace Route Program.

**Suggested Activities:**

**Activity 5:** Implement a multi-threaded TCP server that can handle multiple clients in parallel using pthread_create() and use mutexes and condition variables to manage shared resources.

**Activity 6:** Build a simple Ping and Traceroute utility using raw sockets, and analyze the ICMP packets with tools like Wireshark.

**Simple Network Management**
SNMP Network Management Concepts – SNMPv1 – Management Information – MIB Structure – Object Syntax – Standard MIB 's – MIB–II Groups – SNMPv1 Protocol and Practical Issues – Overview of RMON – Statistics and Collection – Alarms and Filters.

**Suggested Activities:**

**Activity 7:** Design and simulate an SNMP agent that can respond to MIB-II queries, such as system uptime, interface status, and packet counters.

**Activity 8:** Prepare a poster or present a seminar on RMON and how it differs from SNMP in terms of traffic monitoring and remote data collection.

**Network Management Tools & Systems**
System Utilities – Network Status Tools – Traffic monitoring Tools – Network Routing Tools – SNMP Tools – Network Statistics measurement systems – NMS Design – Network Management Systems.

**Suggested Activities:**

**Activity 9:** Use Wireshark or NetFlow to analyze and log network traffic from a local network, and interpret the results (protocols, IPs, throughput).

**Activity 10:** Demonstrate a basic Network Management System (NMS) setup using open-source SNMP tools like Nagios/Zabbix/Cacti, and show how they track device status.

**Course Outcomes:**

1. Develop network-based applications using socket programming and I/O multiplexing techniques.
2. Implement socket options and domain resolution for building robust, protocol-independent applications.
3. Design and manage multi-threaded network applications using raw sockets and system-level synchronization mechanisms.
4. Apply SNMP and RMON protocols for effective network monitoring and management.
5. Utilize tools like Wireshark, NetFlow, and SNMP-based NMS to analyse, troubleshoot, and manage network traffic and performance.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References :**


1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, "UNIX Network Programming: The Sockets Networking API", 3rd Edition, Pearson Education, 2022.
2. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", 8th Edition, Pearson, 2021.
3. Michael J. Donahoo, Kenneth L. Calvert, "TCP/IP Sockets in C: Practical Guide for

Programmers", 2nd Edition, Morgan Kaufmann, 2022.
4. Olivier Bonaventure, "Computer Networking: Principles, Protocols and Practice", Self-Published (Open Textbook), 2020.
5. Thomas A. Limoncelli, Christina J. Hogan, Strata R. Chalup, "The Practice of System and Network Administration", 3rd Edition, Addison-Wesley, 2021.cc

**E-RESOURCES:**

1. https://books.google.com/books/about/UNIX_Network_Programming.html?id=ptSCQgAACAAJ

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Develop network-based applications using socket programming and I/O multiplexing techniques | PO1(3), PO3(3), PO4(2) | 3 | 2 | – |
| **CO2** | Implement socket options and domain resolution for building robust, protocol-independent applications. | PO1(3), PO3(2), PO4(2) | 3 | 2 | – |
| **CO3** | Design and manage multi-threaded network applications using raw sockets and system-level synchronization mechanisms. | PO1(3), PO3(3), PO4(2), PO5(2) | 3 | 3 | – |
| **CO4** | Apply SNMP and RMON protocols for effective network monitoring and management. | PO1(2), PO3(2), PO4(2), PO5(2) | 2 | 3 | – |
| **CO5** | Utilize tools like Wireshark, NetFlow, and SNMP-based NMS to analyse, troubleshoot, and manage network traffic and performance. | PO1(2), PO3(2), PO4(3), PO6(2) | 2 | 3 | 2 |

| MC25004 | Software Project Management | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites -** Software Engineering Concepts

**Course Objective:**
To understand and apply project management methodologies in software development, gain knowledge of risk management tactics and software effort estimation methodologies, and then successfully implement these ideas in practical situations. To equip with practical skills in managing people, organizing teams, and fostering effective team dynamics within the framework of software projects.

**Introduction to Software Project Management**
Defining of Software Development Process - Process - Tailoring the Process - Improving the process discipline - Need for implementing discipline. Software Production Process - Identify the Software Model - Software Process Models: Waterfall Model, Prototyping Model, RAD Model, Incremental Model, Spiral Model, Component Assembly Model - Software Life Cycle.

**Suggested Activities:**

**Activity 1:** Identify and Classify Software Projects
**Activity 2:** Draw and Explain Project Management Life Cycle

**Software Development**
Software Development Team - Three Vital Aspects of Software Project Management - The Team - Meaning of Leadership - Communicating in Harmony - Personality traits - Project Organizations. Project Planning: Top-Down and Bottom-Up Planning - Types of Activity - Project Duration: Schedule Monitoring Tools - Gantt Chart, PERT Chart, Critical Path.

**Suggested Activities:**

**Activity 3:** Create a Project Schedule Using Gantt Chart
**Activity 4:** Draw a Network Diagram and Identify the Critical Path

**Project Review**
Tracking Meetings - Recovery plans - Schedule Work & Escalation Meetings. Project Engineering: Product Requirements - Understanding the Customer Problem to solve - Initial Investigation, Strategies for determining information requirements, Information gathering Tools - Product Objectives.

**Suggested Activities:**

**Activity 5**: Identify and categorize different types of risks in real-world software projects.

**Activity 6:** In pairs or small teams, review each other's code focusing on design integrity and maintainability**.**

**Problem Solving**
Product Specifications - Defining the Final Product - Data Flow Diagram, Data Dictionary, Structured English, Decision Trees, Decision Tables - Feasibility Study. Software Testing:

Test Plan - Development Testing - Verification and Validation - General Testing Methods: White Box and Black Box Testing - Unit Testing - System Integration Testing - Validation Testing - System testing-Agile Testing.

**Suggested Activities:**

**Activity 7:** Teams develop test plans covering unit, integration, system tests using white-box and black-box methods.

**Activity 8:** Identify and classify at least 2 maintenance tasks under the following categories of any software system (e.g., College ERP System, Banking App, or Web-based Email System): Corrective**,** Adaptive**,** Perfective and Preventive

**Software Quality**

Software Quality - Quality Measures - FURPS - Software Quality Assurance - Software Reviews - Format Technical Review (FTR) Formal Approaches to SQA - Software Reliability - Introduction to SQA - The Software Quality Assurance Plan - Formal approaches to SQA - Clean room Methodology

**Suggested Activities:**

**Activity 9:** Students calculate reliability indicators and map functionality/usability/performance to the FURPS framework.

**Activity 10:** Create Product and Process Metrics, Learn how to define and apply software metrics.

**Course Outcomes:**

1. Demonstrate the fundamental concepts of software project management and demonstrate the ability to effectively plan, control, and manage software projects.
2. Apply activity planning and scheduling techniques to identify key project tasks, create project schedules using network models, and optimize overall project duration.
3. Implement risk management principles, including risk identification, assessment, planning, and mitigation, and utilize the PERT technique to manage project scheduling and uncertainty.
4. Analyze software maintenance characteristics, tasks, and challenges, and apply concepts of software reliability, source code metrics, and error estimation to ensure maintainable and dependable software systems.
5. Apply reliability models, testing methodologies, quality frameworks, and software metrics to evaluate and improve software quality.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**

Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Gregory M. Horine, “Project Management Absolute Beginner's Guide”, 5th Edition, Que Publishing, 2022.
2. Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, 6th Edition, McGraw Hill Education, 2020.
3. Robert K. Wysocki, “Effective Project Management: Traditional, Agile, Extreme”, 8th Edition, Wiley, 2023.
4. Pankaj Jalote, “Software Project Management in Practice”, 1st Edition, Pearson Education, 2021.
5. Bennatan, E.M., “Software Project Management: A Practical Approach”, 4th Edition, McGraw Hill, 2022.


**E-Resources:**

1. https://archive.org/details/software-project-management-hughes
2. https://www.smartsheet.com/critical-path-method

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Demonstrate the fundamental concepts of software project management and demonstrate the ability to effectively plan, control, and manage software projects. | PO1(3), PO3(3), PO4(2) | 3 | 2 | - |
| **CO2** | Apply activity planning and scheduling techniques to identify key project tasks, create project schedules using network models, and optimize overall project duration. | PO1(2), PO3(2), PO4(2), PO5(2) | 2 | 2 | - |
| **CO3** | Implement risk management principles, including risk identification, assessment, planning, and mitigation, and utilize the PERT technique to manage project scheduling and uncertainty. | PO1(2), PO3(2), PO4(3), PO6(2) | 2 | 2 | 2 |
| **CO4** | Analyze software maintenance characteristics, tasks, and challenges, and apply concepts of software reliability, source code metrics, and error estimation to ensure maintainable and dependable software systems. | PO1(3), PO2(2), PO4(3) | 2 | 3 | - |
| **CO5** | Apply reliability models, testing methodologies, quality frameworks, and software metrics to evaluate and improve software quality. | PO1(2), PO2(3), PO3(2), PO4(2) | 2 | 3 | - |

| MC25005 | E-Learning | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Pre-requisites:**

1. Explain the components, types, and workflow of e-learning environments.
2. Apply instructional design models to structure learning content and strategies.
3. Create multimedia-rich and interactive e-learning packages using authoring tools.
4. Implement AI-based personalization and analytics within LMS environments.

**Course Objective:**

This course synthesizes the instructional, psychological, technological, and data-driven elements of modern e-learning. It integrates design thinking, multimedia development, platform integration, and AI/analytics to equip CS/IT students to build intelligent, interactive, and effective e-learning solutions.

**Foundations of E-Learning and Design Thinking**

History and Need for E-Learning Types: Synchronous, Asynchronous, Blended Learning–Components of E-Learning Systems–Workflow to Produce and Deliver E-Content–Design Thinking in E-Learning (Empathy → Ideation → Prototype → Test)–Quality Frameworks (SCORM, UDL, ADDIE)–Output: Learning model comparison + design sprint reflection

**Suggested Activities:**

**Activity 1:** Conduct a group seminar on different types of e-learning environments.
**Activity 2:** Collaborative workshop on applying design thinking to a real-world learning scenario.

**Instructional Design & Learning Psychology**

Theories of Learning (Behaviorism, Cognitivism, Constructivism)–Instructional Design Models: ADDIE, SAM, Gagné's Events, Bloom's Taxonomy–Multimedia Learning Principles (Mayer's Theory)– Gamification & Motivation in Learning–Mapping outcomes and assessments (ABCD, Mager)–Output: Storyboard + lesson plan aligned to ID model

**Suggested Activities:**

**Activity 3:** Seminar on comparing major learning theories and their impact on e-learning.
**Activity 4:** Group task to develop an instructional plan using the ADDIE model

**Multimedia & Interactive Content Development**

Authoring Tools: Adobe Captivate, H5P, Canva, Figma–Scriptwriting, Storyboarding, and Visual Flow– Image, Audio, and Video Design for Education–Web–based Interactivity and Usability–Augmented Reality (AR), VR, and GamE-based learning elements–Output: Prototype interactive lesson using a chosen authoring tool

**Suggested Activities:**

**Activity 5:** Hands–on lab: Create a multimedia storyboard using Canva or Figma.
**Activity 6:** Seminar on best practices in creating educational videos and animations.

**Platforms, LMS, and AI in E-Learning**
LMS Architecture: Moodle, Canvas, Custom LMS–xAPI, LTI, SCORM integration–User management, plugin development, webhooks–AI in learning: Recommender Systems, Chatbots (Rasa/GPT), Personalization–Cloud-based deployment, APIs, and LMS analytics dashboards-Output: Mini LMS setup + chatbot or adaptive logic integration

**Suggested Activities:**

**Activity 7:** Seminar on LMS platforms and their extensibility features (Moodle, Canvas).
**Activity 8**: Mini project — Develop a simple chatbot using GPT for a learning topic.

**Evaluation, Analytics & Capstone Project**
Learning Analytics: metrics, dashboards, dropout prediction–A/B Testing, Learner Profiling, UX Feedback–Assessment Strategies (formative, summative, QTI–based MCQs)–Ethics & Privacy in EdTech — AI bias–Final Project: Design, Develop & Demonstrate a Smart E-Learning ModulE-Output: Capstone project + analytics + final pitch/demo

**Suggested Activities:**

**Activity 9:** Capstone Project

**Course Outcomes:**

1. Analyse the pedagogical, technical, and architectural foundations of modern E-learning systems.
2. Apply instructional design principles and learning psychology to create effective digital learning experiences.
3. Design and develop interactive and multimedia–rich learning content using industry–standard tools.
4. Integrate AI, data analytics, and LMS technologies to personalize and enhance learner engagement.
5. Evaluate the quality and impact of E-learning systems through learning analytics and user feedback.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Clark, R. C., & Mayer, R. E., "E-learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning". Wiley, 2016.
2. Horton, W., "E-learning by Design", Pfeiffer, 2011.
3. Ally, M., "Mobile Learning: Transforming the Delivery of Education and Training", Athabasca University Press, 2009.
4. Siemens, G., & Baker, R., "Learning Analytics and Educational Data Mining: Towards Communication and Collaboration", 2012.
5. Mayer, R. E., "Multimedia Learning", 3rd Edition, Cambridge University Press. 2021.


**E-Resources:**

1. https://www.adobe.com/products/captivate.html
2. https://www.classcentral.com/subject/elearning (MOOCs on E-learning)
3. https://learninganalytics.net/
4. https://developer.mozilla.org (for HTML5/JS based interactivity).

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyse the pedagogical, technical, and architectural foundations of modern E-learning systems. | PO1(3), PO3(3), PO4(2) | 3 | 2 | - |
| **CO2** | Apply instructional design principles and learning psychology to create effective digital learning experiences. | PO1(2), PO3(3), PO4(2), PO5(2) | 2 | 2 | - |
| **CO3** | Design and develop interactive and multimedia–rich learning content using industry–standard tools. | PO1(2), PO3(3), PO4(3), PO6(2) | 2 | 2 | 2 |
| **CO4** | Integrate AI, data analytics, and LMS technologies to personalize and enhance learner engagement. | PO1(3), PO2(2), PO4(3) | 2 | 3 | - |
| **CO5** | Evaluate the quality and impact of E-learning systems through learning analytics and user feedback | PO1(3), PO2(3), PO3(2), PO4(2), PO6(2) | 2 | 3 | 2 |

| MC25006 | Accounting And Financial Management | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objective:**

This course aims to help them gain familiarity in the process of capturing, recording and communicating accounting information and to provide in-depth knowledge on successful way of mobilizing and managing finance in a business enterprise.

**Financial Accounting**

Meaning and Scope – Principles – Concepts – Conventions. Journal – Ledger – Trial Balance - Trading Account – Profit and Loss Account – Balance Sheet - Accounting Ratio Analysis– Funds Flow Statement – Cash Flow Statement - Computerized Account.

**Cost and Management Accounting**

Cost Accounting: Meaning. Objectives, Scope and Uses of Cost and Management Accounting — Difference between Cost and Management, Cost and Financial Accounting - Elements of Cost - Cost Volume Profit Analysis - Break Even Analysis.Budgetary Control – Meaning and Types of Budgets – Sales Budget – Production Budget - Cash Budget – Master Budget – Flexible Budgeting – Zero Base Budgeting.

**Financial Management**

Meaning, objectives and scope of financial management. Basis of time value - Finding future value - Discounting and Present value - Future and Present Value of Annuity - Periodicity of Compounding and Discounting. Cost of capital — Measurement of Costs - Equity, Debt and Preference shares — WACC. Capital structure: Determinants — Theories –Optimal capital structure - Leverages — Types.

**Capital Budgeting Decisions**

Objectives and significance - Capital budgeting process - Capital budgeting decisions - Information required for capital budgeting - Techniques and evaluating projects - Risks in capital budgeting - Independent projects and capital rationing - Mutually exclusive projects - Capital budgeting and price level adjustments.

**Dividend Policy and Working Capital Management**

Dividend policy formulation- Dividend Types - Dividend models - Relevance Vs irrelevance of - Theory of relevance. Working capital management- Operating cycle - Working capital policy - Estimation of working capital requirements - sources of working capital financing - Inventory management, Receivables management, Cash management — Baumol, Miller & Orr Model.

**Course Outcomes:**

1. To sketch the financial statements as per accounting concepts, conventions, standards, and to understand and prepare income statement and balance sheet.
2. To understand about the cost and management accounting concepts and tools.
3. To understand the role of finance function and the sources of finance and compute time value of money
4. To apply the capital budgeting techniques and analyze the feasibility of investment proposals.
5. To understand and analyse the dividend decisions and its effect on firm's value and to understand the importance of managing working capital and its components.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |
| **Mandated Activities with marks:**<br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25). | | |
| **Internal Examinations**: TWO tests | | |

**REFERENCES:**


1. Eugene F. Brigham & Louis C. Gapenski, "Financial Management Theory and Practice", Thomson Learning, 10th Edition, 2012.
2. Prasanna Chandra, "Financial Management", Tata McGraw Hill, New Delhi, 2021.
3. R.S.N. Pillai Bagavathi, "Fundamentals of Advanced Accounting", Sultan Chand Publications. 2019.
4. Dr. Ashok Sehgal and Dr. Deepak Sehgal, Accounting for Management, Taxmann Publications (P) Ltd., New Delhi, 2020.
5. T. P. Ghosh, Financial Accounting for Managers, Taxmann Publications (P) Ltd., New Delhi, 4th Edition, 2018.


**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | To sketch the financial statements as per accounting concepts, conventions, standards, and to understand and prepare income statement and balance sheet. | PO1(3), PO3(2), PO5(2) | 2 | - | - |
| **CO2** | To understand about the cost and management accounting concepts and tools. | PO1(3), PO2(2), PO4(2) | 2 | 2 | - |
| **CO3** | To understand the role of finance function and the sources of finance and compute time value of money | PO1(2), PO4(3), PO5(2) | 3 | 2 | - |
| **CO4** | To apply the capital budgeting techniques and analyze the feasibility of investment proposals. | PO1(3), PO3(2), PO4(3) | 3 | 2 | - |
| **CO5** | To understand and analyse the dividend decisions and its effect on firm's value and to understand the importance of managing working capital and its components. | PO1(3), PO2(2), PO4(3), PO6(2) | 2 | 2 | 2 |

| MC25007 | Digital and Mobile Forensics | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Pre- requisites:**

1. Basic knowledge of computer architecture and operating systems
2. Understanding of file systems (FAT, NTFS, ext2/ext3, etc.)
3. Introductory knowledge of cybersecurity principles and threats
4. Basic hands-on skills with computers and mobile devices
5. Exposure to programming or scripting languages

**Course Objective:**

This course provides foundational knowledge of digital and mobile forensics, focusing on principles, tools, and techniques used in the investigation of cyber incidents. It covers procedures for acquiring, analyzing, and preserving digital evidence, while addressing legal and ethical aspects. Students will also gain exposure to mobile forensics and hands-on experience using forensic tools for real-world investigations.

**Foundations of Digital Forensics**

Introduction to Digital Forensics – Chain of custody and evidence handling procedures – Phases of digital forensic investigation – Legal issues: Overview of Indian IT Act and admissibility of digital evidence.

**Suggested Activity:**

**Activity 1:** Case-based discussion on cybercrime scenarios

**Evidence Acquisition and Preservation**

Imaging and hashing: Write blockers, disk cloning – Basics of file systems: FAT, NTFS, ext2/ext3 – Evidence acquisition tools: FTK Imager – EnCase – Autopsy – Memory forensics and volatile data collection – Best practices in evidence preservation

**Suggested Activity:**

**Activity 2:** Hands-on session with imaging tools + flipped classroom session

**Forensic Analysis Techniques**

Data carving and file recovery – Log file and metadata analysis – Email forensics, Internet artifacts – Anti-forensics techniques and detection – Forensic report writing – maintaining chain of custody and proper documentation.

**Suggested Activity:**

**Activity 3:** Poster presentation on analysis techniques

**Mobile Device Forensics**

Overview of Android and iOS architecture – Mobile acquisition methods: logical, physical, and cloud – based – SIM and SD card analysis – Mobile forensic tools: Cellebrite, MOBILedit, Oxygen – App data extraction and analysis.

**Suggested Activity:**

**Activity 4:** Group seminar on mobile forensic tools

**Emerging Trends and Case Studies**
IoT forensics and challenges – Overview of cloud forensics – Blockchain and cryptocurrency forensics – Case studies of real– world investigations – Ethics and future of digital forensics.

**Suggested Activity:**

**Activity 5:** Steganography Puzzle Challenge – Use tools to uncover hidden data within media files

**Course Outcomes:**

1. Analyze the fundamentals and scope of digital and mobile forensics.
2. Apply appropriate methods for acquiring and preserving digital evidence.
3. Analyze digital data using forensic tools to extract relevant information.
4. Conduct forensic investigations on mobile devices using specialized tools.
5. Evaluate emerging trends and ethical considerations in digital forensics.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |
| **Mandated Activities with marks:**<br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25). | | |
| **Internal Examinations**: TWO tests | | |

**References:**


1. Joakim Kävrestad, “Fundamentals of Digital Forensics: Theory, Methods, and Real-Life Applications”, 2nd Edition, Springer, 2020.
2. Greg Gogolin, “Digital Forensics Explained”, 2nd Edition, CRC Press, 2021.
3. Marie-Helen Maras, “Computer Forensics: Cybercriminals, Laws, and Evidence”, 3rd Edition, Jones & Bartlett Learning, 2021.
4. Anchit Bijalwan, “Digital Forensics: Threatscape and Best Practices”, Wiley, 2023.
5. Darren R. Hayes, “A Practical Guide to Digital Forensics Investigations”, Pearson, 2020.


**E- Resources:**

1. NPTEL Course: “Computer Forensics” by Prof. Santosh Biswas
2. DFIR Training (https://www.dfir.training)
3. Autopsy Tool Documentation (https://www.sleuthkit.org/autopsy/docs.php)

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the fundamentals and scope of digital and mobile forensics. | PO1(3), PO3(2), PO4(2) | 3 | 2 | - |
| **CO2** | Apply appropriate methods for acquiring and preserving digital | PO1(3), PO3(3), PO4(2) | 3 | 2 | - |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | evidence. | | | | |
| | **CO3** | Analyze digital data using forensic tools to extract relevant information. | PO1(3), PO2(2), PO4(3) | 3 | 2 | - |
| | **CO4** | Conduct forensic investigations on mobile devices using specialized tools. | PO1(2), PO3(3), PO4(2) | 3 | 3 | - |
| | **CO5** | Evaluate emerging trends and ethical considerations in digital forensics. | PO1(2), PO5(2), PO6(2) | 2 | 2 | 2 |

| MC25008 | Cryptocurrency and Blockchain Techniques | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisite:** Networking and Communication Systems in Modern Computing, Cryptography and Cyber Security

**Course Objective:**
To introduce the major concepts of cryptocurrencies, blockchains, and cryptographic tools. Design, develop and apply smart contracts on blockchain application using Ethereum and emphasis on blockchain privacy and security.

**Foundations of Blockchain and Bitcoin**
History and evolution of blockchain – Blockchain architecture and its functionalities. Types of blockchain: Public, Private, Consortium, Tokenized vs. Tokenless, Layer 1 vs. Layer 2. Bitcoin architecture – Cryptographic keys, addresses, transactions, Merkle trees, SPV. Bitcoin mining – Bitcoin test blockchains: Testnet, Segnet, Regtest – Bitcoin network overview.

**Suggested Activities:**

**Activity 1**: Team Discussion: Usecases in different types of blockchain technologies
**Activity 2**: Quiz: *Cryptography* concepts using real Bitcoin hash examples.

**Consensus Algorithms and Mining**
Introduction to consensus – Design and classification – CFT and BFT algorithms. Decentralized consensus – Transaction verification – Mining process and block creation. Block headers – Hashing race – Chain assembly – Hard Forks vs. Soft Forks – Consensus attacks. Consensus software development and versioning.

**Suggested Activities:**

**Activity 3:** Simulation: Consensus algorithms using Python.
**Activity 4:** Implementation: PoS vs PoW vs BFT techniques

**Smart Contracts and Ethereum Ecosystem**
Introduction to smart contracts – Ricardian contracts – Templates and oracles. Deployment of smart contracts – DAO and advancements – Ethereum architecture. Keys, addresses, EVM, blocks, transactions, accounts, wallets, client software. Protocols supporting Ethereum: Whisper and Swarm.

**Suggested Activities:**

**Activity 5**: Deploy a smart contract on Ethereum Testnet
**Activity 6**: Simulate DAO for the usecase roleplay using DAOStack

**Hyperledger, Tokenization and Enterprise Blockchain**
Overview of Hyperledger projects – Reference architecture – Hyperledger Fabric. Tokenization – Types of tokens, token standards, ERC-20 – Building and deploying tokens. Enterprise blockchain vs. public blockchain – Design and architecture – VMware, Quorum. Setting up a Quorum network using IBFT.

**Suggested Activities:**

**Activity 7:** Mini Project: ERC–20 Token Development and Deployment
**Activity 8:** Team–based simulation: Quorum Network Setup with IBFT

**Bitcoin and Blockchain Security**
Bitcoin security principles – User security best practices. Privacy protocols: Homomorphic encryption, Coinswap, TumbleBit, Dandelion, zkledger, MimbleWimble. Blockchain privacy using zero-knowledge proofs and anonymous signatures. Blockchain security layers – Layer 2 attacks – Threat modeling – Security tools and mechanisms.

**Suggested Activities:**

**Activity 9**: Comparative analysis on privacy protocols
**Activity 10**: Simulating security mechanisms in Bitcoin wallets

**Course Outcomes:**

1. Analyze the fundamental principles of blockchain, cryptocurrencies, and cryptographic techniques.
2. Design various blockchain protocols and evaluate different consensus mechanisms.
3. Develop smart contracts and decentralized applications (DApps) using Ethereum tools such as Solidity and Remix IDE.
4. Analyze real-world blockchain applications in DeFi, NFTs, Web3, and digital identity.
5. Evaluate the security aspects, regulatory challenges, and emerging trends in blockchain technology.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Imran Bashir, “Mastering Blockchain: Inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3”, 4thEdition, Packt Publishing, 2023.
2. Andreas M. Antonopoulos, “Mastering Bitcoin: Programming the Open Blockchain”, 2ndEdition, O’Reilly Media, 2017.
3. Andreas M. Antonopoulos and Gavin Wood, “Mastering Ethereum: Building Smart Contracts and DApps”, 1st Edition, O’Reilly Media, 2018.
4. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, 1st Edition, Princeton University Press, 2016.

5. Roger Wattenhofer, “The Science of the Blockchain”, 1st Edition, CreateSpace Independent Publishing, 2016.

**E- RESOURCES:**

1. https://github.com/bitcoinbook/bitcoinbook
2. https://hyperledger-fabric.readthedocs.io/

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Classify game genres and outline the structure of game development lifecycles and engines. | PO1(3), PO2(2), PO6(2) | 2 | 3 | 2 |
| **CO2** | Apply principles of 2D/3D graphics and animation techniques in designing visual assets. | PO1(3), PO2(3), PO4(2) | 3 | 3 | 2 |
| **CO3** | Simulate game physics and construct basic AI systems using techniques like FSMs and pathfinding. | PO1(3), PO3(3), PO5(2) | 3 | 3 | 2 |
| **CO4** | Design modular game architectures using common game design patterns and scene management techniques. | PO1(3), PO3(2), PO4(2), PO5(2) | 3 | 3 | 2 |
| **CO5** | Evaluate game publishing models, ethical issues, and emerging technologies like AR/VR and cloud gaming. | PO1(3), PO6(3), PO7(3) | 2 | 3 | 3 |

| MC25009 | Game Programming | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisite:** Programming, Data Structures and Algorithms using Python, Enterprise Java Software Development, Full Stack Web Development.

**Course Objective:**
This course aims to introduce students to the foundational principles of game programming, including game design, graphics, physics, and AI. It equips learners with theoretical knowledge of modern game engines and industry practices to conceptualize, design, and evaluate interactive digital games. The course also explores emerging trends, ethical considerations, and business models in the game development ecosystem.

**Introduction to Game Fundamentals and Engines**
**I**ntroduction to Game Programming – Evolution and classification of games – Gnres :action, RPG, simulation, puzzle – Game development life cycle –Overview of Game engine – Unity, Unreal Engine, Godot – Architecture & Scripting models – Game loop and input systems – Game loop structure: updatE-render–input cycle – Real time input handling (keyboard, mouse, mobile touch) Game Documentation**–**Game Design Document (GDD) structure – Creating wireframes and flowcharts for gameplay logic

**Suggested Activities:**

**Activity 1:** Poster Presentation – Compare Unity, Unreal Engine, and Godot in terms of scripting, asset management, rendering capabilities, and platform support.

**Activity 2:** Micro Project – Game Concept Design Create a onE-page design document for a proposed 2D/3D game including genre, story, characters, and gameplay mechanics.

**Graphics and Animation Concepts**
2D Graphics and Coordinator systems – 3D Graphics – Basics– 3D models. Meshes, Materials– Lighting –Camera positioning and Projection, Animation Techniques – Frame-based and skeletal animation, curves –blending, Shaders and Effects – Basics – Visual effects. Graphics Optimization

**Suggested Activities:**

**Activity 3:** Tool Demo– Explore a Sprite Animation Tool
**Activity 4:** Micro Project – Visual Asset Design Brief

**Physics and AI in Games**
Game Physics Concepts – Rigid bodies, gravity, mass, friction –real time simulation Vs arcadE-style physics. Collision Detection – AABB, circle, polygon collision, Bounding boxes and raycasting. Physics engine overview – Box2D, Bullet and PhysX integration. AI Fundamentals in Games – Finite State Machines – Decision trees and scoring systems. Path finding & Navigation – A* algorithm, NavMesh concepts

**Suggested Activities:**

**Activity 5:** Presentation – Physics Engine Comparison
**Activity 6:** Micro Project – Design an AI Behavior Tree

**Game Architecture and Design Patterns**

Game Architecture Overview – Entity-component system model– Layered architecture. Design Pattern for Games – Singleton, Observer, Factory, Component, patterns. Scene and State management – Game states – scene transition logic. Event-driven programming – Data structures in Games

**Suggested Activities:**

**Activity 7:** Presentation – Design Patterns in Game Engines
**Activity 8:** Micro Project – Game Loop Pseudocode

**Industry Practices and Trends**

Game Publishing & Monetization Models – Freemium, premium, ads, in–app purchases, App Store Optimization. Production Pipelines and DevOps for Games – Asset pipelines, Version control (Github), continuous integration. Ethical Issues in Game development – ViolenceE-addiction –privacy – loot boxes. AR/VR and cloud Gaming – Trends and case studies

**Suggested Activities:**

**Activity 9:** Presentation – Game Monetization Models
**Activity 10:** Tool Demo –Micro Project – Business Plan for Indie Game

**Course Outcomes:**

1. Classify game genres and outline the structure of game development lifecycles and engines.
2. Apply principles of 2D/3D graphics and animation techniques in designing visual assets.
3. Simulate game physics and construct basic AI systems using techniques like FSMs and pathfinding.
4. Design modular game architectures using common game design patterns and scene management techniques.
5. Evaluate game publishing models, ethical issues, and emerging technologies like AR/VR and cloud gaming.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:** Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Casey Hardman, “Game Programming Using Unity and C#”, Publisher: Mercury Learning & Information, 2023
2. Jonathan Linowes, “Unity 2023 By Example: Learn about game and virtual reality development by creating five engaging projects”, 5th Edition, Packt Publishing, 2023.
3. Alan Thorn, “Mastering Unity: A Beginner's Guide”, BPB Publications, 2021.
4. Penny de Byl, “Game AI Pro 360: Guide to Movement and Pathfinding”, CRC Press, 2021.

5. Kelsey Gilbert Kreider and Chris Totten, “Designing Game Worlds: Narrative, Architecture, and Interaction”, CRC Press, 2022.

**E- RESOURCES:**

1. https://docs.unity3d.com/Manual/Graphics.html
2. https://docs.godotengine.org/en/stable/tutorials/ai/behavior_tree.html

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Classify game genres and outline the structure of game development lifecycles and engines. | PO1(3), PO2(2), PO5(2) | 2 | 3 | 2 |
| **CO2** | Apply principles of 2D/3D graphics and animation techniques in designing visual assets. | PO1(3), PO3(3), PO5(2) | 3 | 3 | 2 |
| **CO3** | Simulate game physics and construct basic AI systems using techniques like FSMs and pathfinding. | PO1(3), PO2(2), PO4(2), PO5(2) | 3 | 3 | 2 |
| **CO4** | Design modular game architectures using common game design patterns and scene management techniques. | PO1(3), PO3(3), PO4(2) | 3 | 3 | 2 |
| **CO5** | Evaluate game publishing models, ethical issues, and emerging technologies like AR/VR and cloud gaming. | PO6(3), PO7(2), PO12(3) | 1 | 3 | 3 |

| MC25010 | Entrepreneurship Development | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objective:**
The students develop and can systematically apply an entrepreneurial way of thinking that will allow them to identify and create business opportunities that may be commercialized successfully.

**Introduction to Entrepreneurship**
Entrepreneurship – Entrepreneurial mindset – Characteristics of successful entrepreneurs – Types of entrepreneurs – Role of entrepreneurship in economic development – Innovation and creativity – Intrapreneurship – Barriers to entrepreneurship – Entrepreneurship ecosystem – Entrepreneurial process – Start–up India initiative – Women entrepreneurship – Rural entrepreneurship – Social entrepreneurship – Difference between entrepreneur and manager

**Suggested Activities:**

**Activity 1:** Quiz: Basics of entrepreneurship definitions and scope.
**Activity 2:** Flipped classroom: Case study discussion on companies promoting the entrepreneurship.

**Business Planning**
Business plan – Purpose of business planning – Components of a business plan – Feasibility analysis – Market analysis – Financial feasibility – Technical feasibility – Project identification – Project selection – Project appraisal – SWOT analysis – Business model canvas – Value proposition – Revenue models – Elevator pitch – Cost estimation – Risk analysis – PrE-feasibility study

**Suggested Activities:**
**Activity 3:** Seminar: Evaluating Technical Resources for Startups.
**Activity 4:** Poster Presentation: Visual risk matrix for a business idea.

**Entrepreneurial Finance and Funding**
Sources of finance – Angel investors – Venture capital – Seed funding – Crowdfunding – Bootstrapping – Bank loans – Government schemes and subsidies – Financial institutions supporting entrepreneurship – Working capital management – Break–even analysis – Cash flow management – Budgeting for start–ups – Income statement – Balance sheet basics – Investment pitching – Return on investment – Cost-benefit analysis.

**Suggested Activities:**
**Activity 5:** Quiz: Identify bootstrap-friendly business ideas.
**Activity 6:** Field Visit: Visit a local bank to understand start-up loan options.

**Business Operations**
Operations planning – Supply chain management – Inventory management – Quality control – Vendor management – Business scalability – Intellectual property rights – Patents and trademarks – Licensing – Business process automation – Lean start-up methodology – Minimum viable product – Time management – Human resource planning – Organizational structure – Customer relationship management

**Suggested Activities:**

**Activity 7:** Flipped Classroom: Analyze how a startup scaled operations
**Activity 8:** Seminar: Business Licensing Essentials for Entrepreneurs.

**Technical Aspects and Ethics in Entrepreneurship**
Business ethics – Corporate social responsibility – Legal compliancE- Ethical decision making – Global entrepreneurship trends – Technology and entrepreneurship – E-commerce start-ups – Digital marketing for entrepreneurs – Artificial intelligence in business – Blockchain and entrepreneurship – Social media strategies – Mental health for entrepreneurs – Work–life balance – Resilience and leadership – Future of entrepreneurship

**Suggested Activities:**

**Activity 9:** Quiz: Principles of business ethics and compliance.
**Activity 10:** Industrial Visit: Visit a legal consultancy or startup incubator.

**Course Outcomes:**

1. Analyze the fundamentals of entrepreneurship, including the entrepreneurial mindset, characteristics of successful entrepreneurs, and various types of entrepreneurships.
2. Apply tools like the Business Model Canvas and Value Proposition Canvas to design business models with clear revenue streams and customer value.
3. Analyze financial statements including income statements and balance sheets to assess start–up financial health and performance.
4. Analyze the fundamentals of operations planning, supply chain management, and inventory control to ensure efficient business processes.
5. Design digital marketing and social media strategies tailored for entrepreneurial growth**.**

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. S. S. Khanka, “Entrepreneurial development”, S. Chand & Co. Ltd., New Delhi,1999.
2. Kuratko &Hodgetts,”Entrepreneurship-Theory, Process and practices“, Thomson Learning 6thEdition, 2003.
3. Hisrich R D and Peters M P,” Enterpreneurship”, 5thEdition Tata McGraw–Hill,2002.
4. Mathew J Manimala,”Entrepreneurship theory at cross roads, paradigms and praxis” Dream tech, 2ndEdition 2006.
5. Donald F Kuratko,T.V Rao., ”Entrepreneurship: A South Asian perspective”, CengageLearning, 2012.
6. Dr. Vasant Desai, “Small Scale Industries and Entrepreneurship”, HPH, 2006.

**E- Resources:**

1. https://nptel.ac.in/courses/110/106/110106141
2. https://www.bplans.com/business-planning/feasibility-study/

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the fundamentals of entrepreneurship, including the entrepreneurial mindset, characteristics of successful entrepreneurs, and various types of entrepreneurships. | PO1(2), PO6(3), PO8(2), PO9(2), PO10(2) | 2 | 1 | 2 |
| **CO2** | Apply tools like the Business Model Canvas and Value Proposition Canvas to design business models with clear revenue streams and customer value. | PO2(2), PO3(3), PO5(2), PO11(3) | 3 | 2 | 3 |
| **CO3** | Analyze financial statements including income statements and balance sheets to assess start–up financial health and performance. | PO1(3), PO2(2), PO4(2), PO11(2) | 2 | 2 | 2 |
| **CO4** | Analyze the fundamentals of operations planning, supply chain management, and inventory control to ensure efficient business processes. | PO1(2), PO3(2), PO11(2), PO12(2) | 2 | 2 | 2 |
| **CO5** | Design digital marketing and social media strategies tailored for entrepreneurial growth**.** | PO5(3), PO6(2), PO8(2), PO10(3) | 2 | 3 | 3 |

| MC25011 | Generative AI and Prompt Engineering | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisite:** Foundations of Data Science, Machine Learning Techniques

**Course Objective:**

This course enables students to analyze the foundational principles of Transformer models and Generative AI, apply fine-tuning techniques for text and image generation tasks, and evaluate model performance using standard metrics. It also encourages learners to design and implement generative AI solutions using frameworks such as TensorFlow and PyTorch, while critically assessing ethical considerations in model development and deployment.

**Evolution of AI**

Fundamentals of Neural Networks – Evolution from Traditional ML to Deep Learning – Basics of Forward and Backpropagation – Optimization Techniques – Activation Functions – AI Ethics and Responsible AI

**Suggested Activities:**

**Activity 1:** Implement a basic neural network using TensorFlow – Discuss AI ethics challenges

**Foundations of Transformers and Self-Attention Mechanisms**

Introduction to Transformers – Understanding Self-Attention – Positional Encoding – Multi-Head Attention – Key Transformer Architectures: BERT, GPT – Applications of Transformers in AI

**Suggested Activities:**

**Activity 2:**

Fine-tune a simple Transformer model – Analyze self-attention behavior

**Generative AI Techniques and Models**

Generative Adversarial Networks (GANs) – Variational Autoencoders (VAEs) – Diffusion Models – Fine-Tuning for Text and Image Generation – Evaluation Metrics: FID, IS, Perplexity – Ethical Implications in AI-Generated Content

**Activity 3:**Implement GANs for image generation – Explore bias detection in AI-generated text

**Transformer Optimization and Deployment**

Training and Optimization Techniques – Hyperparameter Tuning – Large Language Models (LLMs) – Parameter-Efficient Fine-Tuning (LoRA, QLoRA) – Deployment Strategies: Hugging Face, OpenAI API – Case Studies on Real-World Deployments

**Suggested Activities:**

**Activity 4:**Fine-tune BERT for NLP tasks – Deploy a generative AI model using Hugging Face

**Advanced Topics in Transformers and Security Considerations**

Vision Transformers – Diffusion Models for Image Generation – Adversarial Attacks and AI Security – Robustness in AI Models – Bias and Explainability in AI Systems – Future Trends in Transformer Architectures

**Suggested Activities:**

**Activity 5:**Investigate adversarial attack defenses – Compare Vision Transformers with CNNs

**Practical Applications and Future Research Directions**
Real-World Use Cases of Generative AI – AI in Creative Industries – AI-Assisted Content Creation – Emerging Trends in Multimodal AI – AI in Healthcare and Scientific Research – Ethical Challenges and Regulatory Considerations

**Suggested Activities:**

**Activity 6:**Discuss AI impact on industries – Implement a final project using Generative AI

**Course Outcomes:**

1. Analyze the foundational concepts of deep learning and transformers.
2. Develop hands-on experience with generative models.
3. Implement and fine-tune models using frameworks like TensorFlow and PyTorch.
4. Analyze the impact of generative AI across industries.
5. Optimize Transformer-based architectures for performance and efficiency.
6. Evaluate generative models using appropriate metrics while considering ethical implications.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Ian Goodfellow, YoshuaBengio, Aaron Courville, “Deep Learning”, MIT Press, 1st Edition, 2016, ISBN: 978-0262035613
2. Lewis Tunstall, Leandro von Werra, Thomas Wolf, “Natural Language Processing with Transformers”, O'Reilly Media, 2022, ISBN: 978-1098136796
3. David Foster, “Generative Deep Learning”, O'Reilly Media, 2023, ISBN: 978-1098134181
4. Denis Rothman, “Transformers for Natural Language Processing” Packt Publishing, 2022, ISBN: 978-1803247335

5. Jeremy Howard, Sylvain Gugger "Deep Learning for Coders with fastai and PyTorch", O'Reilly Media, 2020, ISBN: 978-1492045526

**E-RESOURCES:**

1. NPTEL: Introduction to Deep Learning & Generative Models – IIT Madras
2. MOOC: Transformers and NLP – Stanford University
3. SWAYAM: Artificial Intelligence: Basics to Advanced – IISc Bangalore
4. Additional Resources: Hugging Face Tutorials, OpenAI Documentation, TensorFlow &PyTorch Guides

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Introduce foundational concepts of deep learning and transformers. | PO1(3), PO2(2), PO3(3), PO4(1), PO5(2), PO11(1) | 2 | 0 | 1 |
| **CO2** | Develop hands-on experience with generative models. | PO1(3), PO2(3), PO3(3), PO4(3), PO5(3), PO10(1), PO11(2) | 2 | 1 | 2 |
| **CO3** | Implement and fine-tune models using frameworks like TensorFlow and PyTorch. | PO1(2), PO2(2), PO3(3), PO4(3), PO5(2), PO10(3), PO11(2) | 3 | 2 | 1 |
| **CO4** | Optimize Transformer-based architectures for performance and efficiency. | PO1(3), PO2(3), PO3(3), PO4(3), PO5(3), PO10(3), PO11(3) | 3 | 1 | 1 |
| **CO5** | Optimize Transformer-based architectures for performance and efficiency. | PO1(3), PO2(2), PO3(3), PO4(3), PO5(3), PO10(3), PO11(2) | 1 | 1 | 1 |
| **CO6** | Evaluate generative models using appropriate metrics while considering ethical implications. | PO1(2), PO2(1), PO3(3), PO4(3), PO5(2), PO10(2), PO11(1) | 2 | 1 | 1 |

| MC25012 | Wireless Sensor Networks and Personal Area Networks | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites:**

1. Basic knowledge of computer networks and communication models (e.g., OSI, TCP/IP).
2. Familiarity with programming languages such as C or Python.
3. of embedded systems or microcontroller basics.
4. Fundamental concepts of wireless technologies like Bluetooth, ZigBee, or RF.
5. Analytical skills for evaluating protocols, security, and energy efficiency in networks

**Course Objective:**

To provide students with the basic understanding and the fundamental concepts, architecture, and components of Wireless Sensor Networks (WSNs) and Personal Area Networks (PANs). And also to analyze various communication protocols and routing techniques used in WSNs and PANs. Identify energy– efficient techniques and challenges related to power management in wireless sensor devices. Explore security issues, threats, and solutions specific to WSNs and PANs. Examine real–world applications of WSNs and PANs, especially in healthcare and environmental monitoring.

**Introduction to Wireless Sensor Networks**

Definition and characteristics of WSNs– Applications of WSNs (environmental monitoring, military, industrial, healthcare)– Sensor node architecture and hardware components– Types of sensors and actuators– Network topology and deployment strategies

**Suggested Activities:**

**Activity 1:** Short MCQ or reflection quiz (Google Form or LMS) on types of sensors.

**Communication Protocols and Layers in WSN**

Wireless communication fundamentals (RF, ZigBee, Bluetooth, etc.)– MAC protocols for WSN – Routing protocols – Data aggregation and dissemination techniques– Energy efficiency and power management– Addressing and routing challenges in WSNs– Quality of Service (QoS) and reliability– Transport protocols and congestion control– Cross– layer design considerations

**Suggested Activities:**

**Activity 2:** Simulate the network using a WSN simulator and document the key aspects related to communication, protocols, and performance metrics.

**Security in Wireless Sensor Networks**

Security challenges and threats in WSNs– Key management, authentication, and encryption– Secure routing and intrusion detection– Privacy and trust management

**Suggested Activities:**

**Activity 3:** Poster presentation on real–world use cases

**Introduction to Personal Area Networks and Protocols**
Overview and definition of PANs– Differences between WSN and PAN– PAN architecture and components (wearable sensors, implantable sensors)– Standards for PANs (Bluetooth and IEEE 802.15.4 (Zigbee)– Physical and MAC layer protocols specific to PANs– Low– power communication techniques– Data collection, processing, and transmission– Network and application layer protocols

**Suggested Activities:**

**Activity 4:** Assignment on different Standards for PANs

**Modern Network Trends – IoT, Cloud & Secure Networking**
Healthcare monitoring systems– Fitness and wellness applications– Remote patient monitoring– Emergency response and assisted living– Energy harvesting and power management in PANs– Privacy, security, and ethical issues– Interoperability and standardization– Integration with IoT and cloud computing.

**Suggested Activities:**

**Activity 5:** Case Study Topic: "Smart Health Monitoring System for Elderly Care Using Personal Area Networks and IoT"

**Course Outcomes:**

1. Demonstrate the configuration and deployment of Wireless Sensor Networks and Personal Area Networks for specific applications.
2. Evaluate different MAC and routing protocols and select appropriate protocols for specific WSN/PAN applications.
3. Design energy– efficient sensor network systems considering constraints and resource limitations.
4. Apply security mechanisms and protocols to safeguard WSNs and PANs against various attacks.
5. Analyze and propose solutions for data aggregation, communication, and processing challenges in WSNs and PANs.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40%<br>Activities: 15%<br>Internal Test – 1&2: 25% | End Semester Theory Examination: 60% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**

1. Fei Hu, “Wireless Sensor Networks: Principles and Practice” 2nd Edition, CRC Press, 2023

2. Jun Zheng, “Wireless Sensor Networks and Internet of Things: Intelligent Applications and Perspectives”, Abbas Jamalipour, Wiley, 2022.
3. Ian F. Akyildiz, Mehmet Can Vuran , “Wireless Sensor Networks”, Wiley, 2010.
4. Kazem Sohraby, Daniel Minoli, Taieb Znati, “Wireless Sensor Networks: Technology, Protocols, and Applications”, Wiley, 2007.
5. William Stallings, “Wireless Communications and Networks”, Pearson, 2005.

**E-Resources:**

1. Sensor Networks and Applications (Wikipedia)
   https://en.wikipedia.org/wiki/Wireless_sensor_network
2. IEEE Xplore Digital Library – PAN Articles
   https://ieeexplore.ieee.org/Xplore/home.jsp

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Demonstrate the configuration and deployment of Wireless Sensor Networks and Personal Area Networks for specific applications. | PO1(3), PO2(2), PO5(3) | 3 | 3 | 2 |
| **CO2** | Evaluate different MAC and routing protocols and select appropriate protocols for specific WSN/PAN applications. | PO1(3), PO2(3), PO3(2), PO4(2) | 3 | 3 | 2 |
| **CO3** | Design energy– efficient sensor network systems considering constraints and resource limitations. | PO1(3), PO3(3), PO5(2) | 3 | 3 | 3 |
| **CO4** | Apply security mechanisms and protocols to safeguard WSNs and PANs against various attacks. | PO2(2), PO6(3) | 2 | 2 | 3 |
| **CO5** | Analyze and propose solutions for data aggregation, communication, and processing challenges in WSNs and PANs. | PO1(3), PO2(2), PO4(2) | 3 | 3 | 2 |

| MC25013 | Fog and Edge Computing | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites:**

1. Basic knowledge of computer networks, including protocols, IP addressing, and communication models.
2. Understanding of cloud computing concepts, such as virtualization, distributed systems, and service models (IaaS, PaaS, SaaS).
3. Familiarity with IoT fundamentals, including sensor networks and data flow in IoT systems.
4. Basic programming skills, especially in Python, C/C++, or Java for devicE- level computing.
5. Awareness of real– time systems and latency– sensitive applications.
6. Knowledge of operating systems and resource management in constrained devices (e.g., Raspberry Pi, microcontrollers).

**Course Objective:**

To provide a comprehensive understanding of Fog Computing as an extension of cloud services toward the edge of the network. It covers core concepts, architecture, optimization techniques, and real– time applications in domains such as healthcare, surveillance, and smart transportation. The course also addresses formal modeling, performance metrics, testing perspectives, and legal aspects, enabling learners to design, evaluate, and deploy intelligent, secure, and efficient fog– based IoT systems.

**Internet of Things (IoT) and New Computing Paradigms**

Introduction-Relevant Technologies –Fog and Edge Computing– Hierarchy of Fog and Edge Computing-Business Models-X as a Service- Support Service-Application Service-Opportunities and Challenges-Out-of-Box Experience-Open Platforms-System Management.

**Suggested Activities:**

**Activity 1**: Poster presentation: Hierarchy of Fog and Edge Computing.
**Activity 2**: Debate on Opportunities and Challenges

**Challenges in Federating Edge Resources**

Introduction – Networking Challenge-Methodology – Integrated C2F2T Literature by Modeling Technique – Integrated C2F2T Literature by Use – Case Scenarios – Integrated C2F2T Literature by Metrics – Threads – Standards

**Suggested Activities:**

**Activity 3:** C2F2T UsE- Case Mapping Workshop
**Activity 4:** C2F2T Metrics and Standards Roleplay Debate

**Orchestration of Network Slices in Fog, Edge, and Clouds**

Introduction – Background – Network Slicing – Network Slicing in SoftwarE- Defined Clouds– Network Slicing Management in Edge and Fog – Internet of Vehicles (IoV): Architecture, Protocols and Seven– layer security model architecture for Internet of Vehicles – IoV: Network Models, Challenges and future aspects

**Suggested Activities:**

**Activity 5:** Design Your Own IoV Security Stack
**Activity 6:** Network Slicing Strategy Simulation.

**Optimization Problems in Fog and Edge Computing**

Preliminaries – The Case for Optimization in Fog Computing– Formal Modeling Framework for Fog Computing – Metrics – Further Quality Attributes – Optimization Opportunities along the Fog Architecture – Optimization Opportunities along the Service Life Cycle – Toward a Taxonomy of Optimization Problems in Fog Computing

**Suggested Activities:**

**Activity 7:** Presentation – Fog Optimization Taxonomy Builder
**Activity 8:** Micro Project – Metric– Based Optimization.

**Applications of Fog and Edge Computing**

Exploiting Fog Computing in Health Monitoring– Smart Surveillance Video Stream Processing at the Edge for Real – Time Human Objects Tracking– Fog Computing Model for Evolving Smart Transportation Applications – Testing Perspectives of Fog – Based IoT Applications – Legal Aspects of Operating IoT Applications in the Fog.

**Suggested Activities:**

**Activity 9:** Fog Application Blueprint – Sector– wise Design Challenge
**Activity 10:** Flipped Classroom for IoT in Fog Computing

**Course Outcomes:**

1. Apply the architecture, characteristics, and evolution of Fog Computing as an extension of cloud computing.
2. Analyze various optimization opportunities, modeling frameworks, and performance metrics in fog– based environments.
3. Design fog– based solutions for real– time applications in healthcare, surveillance, and transportation domains.
4. Evaluate testing methodologies, deployment challenges, and quality attributes of Fog– IoT integrated systems.
5. Interpret and apply legal, ethical, and regulatory considerations in the operation of IoT applications within the fog paradigm.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Rajkumar Buyya and Satish Narayana Srirama, “Fog and Edge computing: Principles and Paradigms”, 2019, 1st Edition, John Wiley & Sons, USA.
2. Arshdeep Bahga and Vijay Madisetti, “Cloud computing: A Hands-On Approach”, 2014, 2nd Edition, CreateSpace Independent Publishing Platform, USA.
3. Ovidiu Vermesan, Peter Friess, “Internet of Things –From Research and Innovation to Market Deployment”, 2014, 1st Edition, River Publishers, India.
4. Jie Cao, Quan Zhang, Weisong Shi,”Edge Computing: Models, Technologies and Applications”, Springer, 2020.


**E-Resources:**

1. Microsoft Azure – What is Edge Computing?-https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/
2. Cisco – Fog Computing Overview https://www.cisco.com/c/en/us/solutions/internet-of-things/fog-computing.html

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Apply the architecture, characteristics, and evolution of Fog Computing as an extension of cloud computing. | PO1(3), PO2(2), PO5(2) | 3 | 2 | - |
| **CO2** | Analyze various optimization opportunities, modeling frameworks, and performance metrics in fog– based environments. | PO1(3), PO2(3), PO4(2), PO5(3) | 3 | 2 | 2 |
| **CO3** | Design fog– based solutions for real– time applications in healthcare, surveillance, and transportation domains. | PO1(3), PO2(2), PO3(3), PO4(3) | 3 | 3 | 2 |
| **CO4** | Evaluate testing methodologies, deployment challenges, and quality attributes of Fog– IoT integrated systems. | PO1(3), PO4(3), PO6(2) | 2 | 2 | 2 |
| **CO5** | Interpret and apply legal, ethical, and regulatory considerations in the operation of IoT applications within the fog paradigm. | PO6(3), PO5(2), PO1(2) | 1 | 2 | - |

| MC25014 | Internet of Things | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Prerequisites:**

1. Basic electronics knowledge, including sensors, actuators, and microcontrollers.
2. Understanding of computer networks, especially TCP/IP, HTTP, and MQTT protocols.
3. Familiarity with embedded systems, such as Arduino, Raspberry Pi, or ESP32.
4. Basic programming skills, preferably in C/C++, Python, or JavaScript.
5. Knowledge of cloud platforms and data handling, such as AWS IoT, Azure IoT, or ThingSpeak.
6. Introductory understanding of cybersecurity concepts for securing IoT devices.

**Course Objective:**

This course introduces the fundamentals of the Internet of Things (IoT), including architecture, communication protocols, and hardware platforms. Students will gain practical experience in designing and developing IoT systems using sensors, actuators, and embedded boards with real– time data communication and cloud integration.

**Introduction to IoT**

Definition– characteristics, and scope of IoT. Evolution of IoT– applications in various domains (smart home, healthcare, agriculture). IoT ecosystem and enabling technologies– IoT Architecture: Perception layer– Network layer– Application layer. Edge computing and Cloud computing in IoT.

**Suggested Activities:**

**Activity 1:** Seminar: Evolution and current trends in IoT
**Activity 2:** Quiz on IoT layers and architecture

**IoT Communication and Networking**

IoT Communication Models and APIs– Protocols: MQTT– CoAP– AMQP– HTTP. Wireless Technologies: Wi– Fi– Bluetooth– ZigbeE- LoRa– NFC– 5G. IP addressing: IPv4 and IPv6– addressing schemes in IoT. Sensor Networks– data acquisition and transmission models.

**Suggested Activities:**

**Activity 3:** Poster presentation: IoT protocols comparison
**Activity 4:** Hands– on: Simulate devicE- to– device communication using MQTT

**IoT Hardware and Devices**

Overview of 8051 microcontroller-Arduino– Raspberry Pi. Interfacing of sensors and actuators (temperature, humidity, motion, light, etc.). Real– time data acquisition and processing– Analog and digital sensor handling– Basics of embedded C.

**Suggested Activities:**

**Activity 5:** Mini project: Build a temperature monitoring system using Arduino and IoT cloud
**Activity 6:** Project– based Learning: Design an IoT– based Smart Parking prototype

**Data Management and Cloud Platforms**
IoT data storage and analytics– Open source IoT platforms: ThingsBoard – NodE-RED– Cloud integration: AWS IoT– Azure IoT Hub– Google Cloud IoT– Data visualization tools and dashboards. Security and privacy in IoT systems.

**Suggested Activities:**

**Activity 7:** Mini project: Send sensor data to ThingsBoard and display on dashboard.

**Activity 8:** Video demo + discussion: Setting up IoT cloud dashboards (AWS/GCP).

**IoT Applications and Trends**
Complete Design of Embedded Systems – Development of IoT Applications – Home Automation – Smart Agriculture – Smart Cities – Smart Healthcare. Challenges in deployment and scalability– Research directions and opportunities in IoT.

**Suggested Activities:**

**Activity 9:** Project expo: Build a mini smart system (e.g., Smart Dustbin, Home Temp Monitor).

**Activity 10:** Research paper review: Emerging trends like Edge AI or Digital Twins.

**Course Outcomes:**

1. Apply IoT principles to design solutions using appropriate architectures and enabling technologies for real– world problems.
2. Analyze and apply IoT communication protocols and networking concepts.
3. Interface sensors, actuators with microcontrollers to build functional IoT systems.
4. Design and deploy real– time IoT applications using embedded platforms and cloud services.
5. Demonstrate collaborative project development and presentation of IoT– based solutions.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 40% | End Semester Theory Examination: 60% |
| | Activities: 15%<br>Internal Test – 1&2: 25% | |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**References:**


1. Arshdeep Bahga and Vijay Madisetti, “Internet of Things: A Hands– On Approach”, 2nd Edition, VPT, 2021.
2. Pethuru Raj, Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases”, CRC Press, 2020.
3. Sunil Cheruvu et al., “Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment”, Apress, 2020.
4. Shweta Jaiswal, “Internet of Things (IoT): Architecture, Technology, and Applications”, Wiley, 2022.
5. Yashavant Kanetkar, “Exploring C for IoT Development”, BPB Publications, 2022.


**E-Resources:**

1. NPTEL: https://nptel.ac.in/courses/106105166
2. Cisco IoT Learning Path: https://www.netacad.com/courses/iot
3. IoT Development on ThingsBoard: https://thingsboard.io/docs
4. Arduino Official: https://www.arduino.cc

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Apply IoT principles to design solutions using appropriate architectures and enabling technologies for real– world problems. | PO1(3), PO2(3), PO3(2) | 3 | 3 | 2 |
| **CO2** | Analyze and apply IoT communication protocols and networking concepts. | PO1(3), PO2(3), PO4(2) | 3 | 3 | - |
| **CO3** | Interface sensors, actuators with microcontrollers to build functional IoT systems. | PO1(3), PO3(3), PO5(2) | 3 | 2 | 2 |
| **CO4** | Design and deploy real– time IoT applications using embedded platforms and cloud services. | PO2(3), PO3(3), PO5(3) | 3 | 3 | 3 |
| **CO5** | Design and deploy real– time IoT applications using embedded platforms and cloud services. | PO4(2), PO6(3) | 2 | 2 | - |

| MC25015 | Deep Learning Techniques | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Prerequisites:**

1. Strong foundation in linear algebra and calculus (e.g., matrices, vectors, gradients).
2. Understanding of probability and statistics, especially for modeling uncertainty and distributions.
3. Basic machine learning knowledge, including supervised/unsupervised learning and common algorithms.
4. Proficiency in Python programming, especially with libraries like NumPy, Pandas, and Matplotlib.
5. Familiarity with neural networks, activation functions, and backpropagation.
6. Hands– on experience with ML frameworks like TensorFlow or PyTorch (basic level)..

**Course Objective:**

This course introduces the foundational and advanced concepts of neural networks and deep learning. It covers the architecture, training, and optimization of various neural models including CNNs, RNNs, and LSTMs. Students will gain hands-on experience with deep learning techniques such as transfer learning and generative models. The objective is to enable students to build, train, and evaluate intelligent systems for real-world applications

**Fundamentals of Neural Networks**

Introduction to artificial neurons – McCulloch-Pitts model, perceptron, and activation functions.
Learning paradigms – supervised, unsupervised, and reinforcement learning. Architecture of neural networks – single-layer, multi-layer, and feedforward networks. Training of neural networks – error correction, backpropagation, and gradient descent.

**Suggested Activities:**

**Activity 1:** Quiz on Perceptron and Backpropagation.

**Practical Activities:**

1. Implement a perceptron model to classify linearly separable data using supervised learning.
2. Build a multi-layer feedforward neural network and train it using the backpropagation algorithm.

**Deep Neural Networks**

Introduction to deep learning – concepts, need, and advantages over shallow networks. Deep neural network architectures – input, hidden, and output layers; role of activation functions. Backpropagation in deep networks – forward pass, backward pass, and weight updates. Challenges in deep learning – vanishing gradients, overfitting, and strategies to overcome them.

**Suggested Activities:**

**Activity 2:** Poster presentation on Activation and optimization algorithms.

**Practical Activities:**

1. Visualize various activation functions and analyze their behavior and impact on learning.
2. Train a deep neural network on a standard dataset and evaluate its performance.

**Convolutional Neural Network (CNN) Architectures**

Introduction to CNNs – motivation, basic structure, and comparison with fully connected networks. Convolutional layers, pooling layers, activation functions, and flattening – their roles and functions. Popular CNN architectures – LeNet, AlexNet, GoogLeNet (overview and Comparison).Training CNNs – parameter tuning, transfer learning, and use of pretrained models.

**Suggested Activities:**

**Activity 3:** CNN Layer Walkthrough – Think, Pair, Share"

**Practical Activities:**

1. Design and implement a Convolutional Neural Network (CNN) for basic image classification.
2. Perform image classification using pre-trained CNN architectures through transfer learning.
3.

**Transfer Learning and RNN**

Introduction to transfer learning – concept, benefits, and use of pretrained models. Fine-tuning and feature extraction using popular models like VGG, ResNet, and BERT. Introduction to RNNs – architecture, sequential data processing, and vanishing gradient problem.

Variants of RNN – LSTM and GRU; applications in text, speech, and time-series data.

**Suggested Activities:**

**Activity 4:** Mini Project with transfer learning.

**Practical Activities:**

1. Build a basic Recurrent Neural Network (RNN) to handle sequential data such as text or time series.
2. Implement LSTM and GRU networks to solve long-term dependency problems in sequential datasets.

**Generative Models**

Introduction to generative models – concepts, applications, and types (explicit vs. implicit).

Autoencoders and Variational Autoencoders (VAE) – architecture and latent space learning. Generative Adversarial Networks (GANs) – components, training process, and challenges. Applications of generative models in image synthesis, data augmentation, and creative AI.

**Suggested Activities:**

**Activity 5:** Visualizing and Comparing Generated Images from Autoencoders and GANs

**Practical Activities:**

1. Develop an autoencoder to perform dimensionality reduction and reconstruct input data.
2. Train a simple Generative Adversarial Network (GAN) to generate synthetic images and visualize the results.

**Course Outcomes:**

1. Analyze the structure and functionality of artificial neurons, perceptrons, and learning paradigms in neural networks.
2. Design and optimize deep neural networks using appropriate training strategies and overcome challenges like vanishing gradients.
3. Develop and evaluate CNN architectures for solving complex image classification problems.
4. Adapt and implement transfer learning and recurrent neural networks for advanced sequence modelling tasks.
5. Construct and critically assess generative models such as Autoencoders, VAEs, and GANs for creative AI applications.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>i. Activities: 15%<br>ii. Internal Theory Examinations: 30%<br>iii. Internal Laboratory Examinations: 15% | End Semester Theory Examination: 40% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**REFERENCES:**


1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2016.
2. Simon Haykin, "Neural networks and Learning Machines", Third Edition, Pearson, 2016.
3. François Chollet," Deep Learning with Python", Manning, 2nd Edition, 2021.
4. Josh Patterson & Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly, 2017.
5. A. Géron, "Hands– On Machine Learning with Scikit– Learn, Keras, and TensorFlow", O'Reilly, 3rd Edition, 2022.
6. Charu Aggarwal, "Neural Networks and Deep Learning", Springer, 2018.

**E-RESOURCES:**

1. https://www.tensorflow.org/tutorials
2. https://onlinecourses.nptel.ac.in/noc21_cs76/preview
3. https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB– 3pi
4. https://keras.io
5. Edutech:https://www.edutechlearning.com/coursedesc/computer-science/lab-courses/deep-learning-lab-course

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the structure and functionality of artificial neurons, perceptrons, and learning paradigms in neural networks. | PO1(3), PO2(2), PO3(2) | 3 | – | – |
| **CO2** | Design and optimize deep neural networks using appropriate training strategies and overcome challenges like vanishing gradients. | PO1(3), PO2(3), PO3(3), PO5(2) | 3 | – | – |
| **CO3** | Develop and evaluate CNN architectures for solving complex image classification problems. | PO1(3), PO2(3), PO3(3), PO4(2), PO5(2) | 3 | – | – |
| **CO4** | Adapt and implement transfer learning and recurrent neural networks for advanced sequence modeling tasks. | PO1(3), PO3(3), PO4(3), PO5(2), PO6(2) | 3 | – | – |
| **CO5** | Construct and critically assess generative models such as Autoencoders, VAEs, and GANs for creative AI applications. | PO1(3), PO2(3), PO3(3), PO6(2) | 3 | – | 2 |

| MC25016 | Artificial Intelligence | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Prerequisites:**

1. Proficient understanding of discrete mathematics, especially in logical inference, set theory, and probability distributions.
2. Strong foundation in data structures and algorithm design, including recursion, graphs, and search strategies.
3. Ability to implement algorithmic solutions using Python or a comparable high-level programming language, with exposure to libraries relevant to AI (e.g., NumPy, scikit-learn).

**Course Objective:**

To impart foundational knowledge of Artificial Intelligence (AI), including its principles, techniques, and historical evolution. It enables learners to assess the applicability, strengths, and limitations of various knowledge representation schemes, problem-solving strategies, and learning methods in addressing real-world engineering problems. Furthermore, the course focuses on developing intelligent systems by integrating appropriate AI techniques to solve concrete computational challenges

**Introduction to Artificial Intelligence**

Foundations of AI – history, definition, and scope; AI as the study of intelligent agents. The state of the art – applications, current trends, and research challenges. Rational agents – structure, performance measures, and PEAS descriptions. Environment types – fully/partially observable, deterministic/stochastic, episodic/sequential.

**Suggested Activities:**

**Activity 1:** Timeline Creation – Evolution of AI.
**Activity 2:** Role-play– AI Agents.

**Practical Activities:**

1. Design an intelligent agent that makes decisions based on performance measures and environmental conditions.

**Problem Solving based on Searching**

Introduction to Problem Solving by searching Methods– State Space Search-Uninformed Search Methods – Uniform Cost Search, Breadth First Search– Depth First Search– Depth– limited search, Iterative deepening depth first, Bidirectional Search-Informed Search Methods– Best First Search-A*,SMA* Search.

**Suggested Activities:**

**Activity 3:** Quiz on Uninformed Searches.
**Activity 4:** Role-play – Token vs Traditional Payment.

**Practical Activities:**

1. Implement basic uninformed search algorithms such as BFS, DFS, and Uniform Cost Search for problem-solving tasks.
2. Apply informed search techniques like Best-First Search and A* to navigate through heuristic-based problems.

3. Simulate iterative deepening and bidirectional search algorithms and compare their efficiencies.

**Local Search and Adversarial Search**

Local Search algorithms – Hill– climbing search-Simulated annealing- Genetic Algorithm-Adversarial Search-Game Trees and Minimax Evaluation- Elementary two-player game, tic– tac– toe-Minimax with Alpha– Beta Pruning.

**Suggested Activities:**

**Activity 5:** Case Study – Alpha– Beta Pruning.
**Activity 6:** Hands– on Task – Minimax Evaluation.

**Practical Activities:**

1. Solve constraint satisfaction or optimization problems using local search methods like hill climbing and simulated annealing.
2. Create a two-player game using the Minimax algorithm with Alpha–Beta pruning to demonstrate adversarial search.

**Logical Reasoning**

Introduction to propositional and predicate logic – logical connectives, truth tables, and normal forms. Rules of inference – direct and indirect proofs, contradiction, and logical equivalence. Automated reasoning – resolution, unification, forward and backward chaining. Applications in AI – knowledge representation, monotonic and non-monotonic reasoning.

**Suggested Activities:**

**Activity 7:** Presentation – Forward and backward chaining.
**Activity 8:** Case Study on Resolution applications.

**Practical Activities:**

1. Construct truth tables and use propositional logic to verify logical equivalence and infer conclusions.
2. Use predicate logic and resolution techniques for automated reasoning in a knowledge-based system.

**Reasoning Under Uncertainty in AI**

Introduction to uncertainty – sources of uncertainty, need for probabilistic reasoning. Bayesian networks – structure, conditional probability, inference in Bayesian networks. Probabilistic reasoning over time – hidden Markov models, dynamic Bayesian networks. Decision making under uncertainty – utility theory, value of information, and decision networks.

**Suggested Activities:**

**Activity 9:** Mind Mapping – Planning.
**Activity 10:** Quiz on Bayesian Belief Network.

**Practical Activities:**

1. Build a Bayesian network for probabilistic reasoning and perform inference based on given evidence.
2. Model and evaluate decisions under uncertainty using utility theory and decision networks.

**Course Outcomes:**

1. Examine the foundational concepts and evolution of AI to differentiate between various types of rational agents and their use cases.
2. Apply uninformed and informed search algorithms for problem solving in state-space environments.
3. Implement local search techniques and adversarial game strategies using algorithms like Minimax and Alpha–Beta pruning.
4. Use propositional and predicate logic for reasoning and apply automated inference techniques.
5. Analyze and apply probabilistic models like Bayesian networks and decision theory for reasoning under uncertainty.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>i. Activities: 15%<br>ii. Internal Theory Examinations: 30%<br>iii. Internal Laboratory Examinations: 15% | End Semester Theory Examination: 40% |

**Mandated Activities with marks:**
Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25).

**Internal Examinations**: TWO tests

**REFERENCES:**


1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2021.
2. Elaine Rich, Kevin Knight, Shivashankar B. Nair, "Artificial Intelligence", 3rd Edition, Tata McGraw– Hill, 2017.
3. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw– Hill Education, 2013.


**E-RESOURCES:**

1. NPTEL Course: Artificial Intelligence by Prof. Mausam (IIT Delhi) – nptel.ac.in/courses/106102220.
2. AI Open-Source Tools & Datasets: https://www.kaggle.com/
3. Google AI Blog: https://ai.googleblog.com
4. Edutech: https://www.edutechai.net/

**CO PO Mapping:**

| CO No. | CO Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Examine the foundational concepts and evolution of AI to differentiate between various types of rational agents and their use cases. | PO1(3), PO2(2) | 3 | – | – |
| **CO2** | Apply uninformed and informed search algorithms for problem solving in state-space environments. | PO1(3), PO3(3), PO5(2) | 3 | 2 | – |
| **CO3** | Implement local search techniques and adversarial game strategies using algorithms like Minimax and Alpha–Beta pruning. | PO1(3), PO3(3), PO4(2) | 3 | 2 | – |
| **CO4** | Use propositional and predicate logic for reasoning and apply automated inference techniques. | PO1(3), PO2(2), PO5(2) | 3 | – | – |
| **CO5** | Analyze and apply probabilistic models like Bayesian networks and decision theory for reasoning under uncertainty. | PO1(3), PO3(3), PO6(2) | 3 | – | 2 |

| MC25017 | DevOps | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Prerequisites:**

1. Proficiency in at least one programming language (e.g., Python, Java, or C++) to follow scripting and automation tasks.
2. Exposure to Linux/Unix commands and file systems, as many DevOps tools operate in Linux environments.
3. Prior knowledge of software development stages and lifecycle models like Waterfall and Agile.
4. Familiarity with version control concepts (e.g., Git, repositories, commits, branches) to effectively engage in collaborative development and CI workflows.
5. Awareness of cloud service models (IaaS, PaaS, SaaS) and platforms (AWS, Azure, GCP) is beneficial for infrastructure and DevSecOps components.

**Course Objective:**

To equip students with comprehensive knowledge and practical skills in DevOps practices, including version control, CI/CD pipelines, infrastructure as code, containerization, monitoring, and security integration into DevOps pipelines using real– world tools and cloud platforms.

**Introduction to DevOps**

Introduction to DevOps: Overview of Software Development Lifecycle (SDLC) – Traditional vs. Agile vs. DevOps – DevOps: Definition, Need, and Benefits – DevOps Principles and Practices – DevOps Lifecycle and Workflow – DevOps Tools Landscape Overview – Culture, Collaboration & Communication in DevOps – Key Roles in a DevOps Team.

**Suggested Activities:**

**Activity 1:** SDLC Comparison RolE- Play.
**Activity 2:** DevOps Lifecycle Simulation.

**Practical Activities:**

1. Compare Traditional vs Agile vs DevOps workflows using a sample project timeline.
2. Map a real– world use case to the DevOps lifecycle (with stages like Plan → Code → Build → Test → Release → Deploy → Monitor).
3. Create a DevOps workflow diagram using draw.io or Lucid chart and explain each stage.

**Version Control and Continuous Integration**

Introduction to Version Control Systems (VCS) – Git Basics: Repositories, Branches, Merges – GitHub/GitLab/Bitbucket Overview – Introduction to CI (Continuous Integration) – CI Concepts and Best Practices – Jenkins: Installation and Configuration – Creating and Managing Jenkins Jobs – Integration of Git with Jenkins – CI Pipeline Design and Implementation.

**Suggested Activities:**

**Activity 3:** Git and GitHub Hands– On Practice.
**Activity 4:** CI Pipeline Setup with Jenkins.

**Practical Activities:**

1. Push a local repository to GitHub/GitLab. Perform pull requests and branch merging.
2. Create a Jenkins pipeline (Declarative/Scripted) to build and test a simple Java/Python app.
3. Integrate GitHub repository with Jenkins CI. Automate build triggers on code push.

**Configuration Management and Containerization**

Infrastructure as Code – Concept and Benefits – Configuration Management Tools Overview - Ansible, Chef, Puppet – Introduction to Docker – Docker Architecture and Components – Docker Images, Containers, Volumes, and Networks – Docker file: Writing and Building Custom Images – Docker Compose for Multi– container Applications – Container Orchestration Introduction- Kubernetes Basics.

**Suggested Activities:**

**Activity 5:** Flipped Classroom on “Exploring Docker & IaC Tools”.
**Activity 6:** Quiz on “Demystifying Containerization and Orchestration”.

**Practical Activities:**

1. Write a basic Ansible playbook to install Apache/Nginx on a remote server (localhost via SSH).
2. Install Docker and run basic commands: docker run, docker ps, docker exec, docker stop
3. Build a Docker image using a custom Docker file and run the container.

**Continuous Delivery, Deployment, and Monitoring**

Continuous Delivery vs Continuous Deployment – CD Pipeline Setup and Implementation – Introduction to GitOps – Deployment Strategies: BluE- Green, Canary, Rolling Updates – Introduction to Monitoring and Logging – Tools: Prometheus, Grafana, ELK Stack – Alerts and Dashboards – Incident Management and Feedback Loop.

**Suggested Activities:**

**Activity 7:** Flipped Classroom Discussion on “GitOps and Deployment Strategies”.
**Activity 8:** Case Study Analysis and Presentation on any topic in unit.

**Practical Activities:**

1. Setup a full CI/CD pipeline using Jenkins with a deploy step (use a dummy deployment to a folder or local server).
2. Demonstrate blue– green or canary deployment using two versions of a simple web app (manual simulation or scripted).

**Cloud Platforms and DevSecOps**

Overview of Cloud Computing Models - IaaS, PaaS, SaaS– Introduction to AWS/Azure/GCP Basics – Setting Up Infrastructure Using Terraform (IaC) – Serverless and Cloud– Native DevOps – Introduction to DevSecOps – Need and Concepts – Security Practices in CI/CD Pipelines – Secrets Management and Vulnerability Scanning – Final Case Study: End– to– End DevOps Implementation.

**Suggested Activities:**

**Activity 9:** RolE- Playon DevSecOps Incident Response Simulation.
**Activity 10:** Using the official pricing calculators from AWS, Azure, or GCP, they estimate the monthly cost.Then discuss ways to optimize cost without compromising performance or security.

**Practical Activities:**

1. Write a basic Terraform script to provision AWS/GCP/Azure EC2 instance (can simulate with local provider if cloud not available).
2. Final Mini Project: Build a full DevOps pipeline for a sample app (Code → Git → Jenkins → Docker → Deploy → Monitor) simulating real– time workflow

.

**Course Outcomes:**

1. Analyze the differences between traditional, Agile, and DevOps methodologies and formulate the need for DevOps in modern software development.
2. Design and implement automated version control and continuous integration pipelines using Git and Jenkins for real-time software delivery.
3. Construct scalable and reproducible infrastructure using Infrastructure as Code (IaC) tools and evaluate containerization strategies using Docker and Kubernetes.
4. Develop and optimize continuous delivery and deployment workflows by integrating monitoring, logging, and feedback mechanisms using tools like Prometheus and ELK.
5. Assess cloud-native DevOps and integrate security controls into CI/CD pipelines by applying DevSecOps principles and cost-effective cloud deployment strategies.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>i. Activities: 15%<br>ii. Internal Theory Examinations: 30%<br>iii. Internal Laboratory Examinations: 15% | End Semester Theory Examination: 40% |
| **Mandated Activities with marks:**<br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25). | | |
| **Internal Examinations**: TWO tests | | |

**References:**


1. Kim, G., Humble, J., Debois, P., & Willis, J., “The DevOps Handbook: How to Create World–Class Agility, Reliability, & Security in Technology Organizations”, 2nd Edition, IT Revolution Press, 2021.
2. Krief, M., “Learning DevOps: Continuously Deliver Better Software”, Packt Publishing, 2020.
3. Poulton, N., “Docker Deep Dive”, 2025 Edition, Shelter Island, 2025.
4. Forsgren, N., Humble, J., & Kim, G., “Accelerate: The Science of Lean Software and DevOps”, IT Revolution Press, 2018.
5. Brikman, Y., “Terraform: Up and Running – Writing Infrastructure as Code”, 2nd Edition, O’Reilly Media, 2019.

**E– Resources:**

1. NPTEL: DevOps by Prof. Rajkumar Buyya.
2. AWS Free Tier Labs.
3. Docker Playground (https://labs.play– with– docker.com).
4. Kubernetes Bootcamp by Google.
5. GitOps Practices by Weaveworks.

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the differences between traditional, Agile, and DevOps methodologies and formulate the need for DevOps in modern software development. | PO1(3), PO2(2), PO6(2) | 2 | 2 | - |
| **CO2** | Design and implement automated version control and continuous integration pipelines using Git and Jenkins for real-time software delivery. | PO2(3), PO3(3), PO5(2) | 3 | 2 | - |
| **CO3** | Construct scalable and reproducible infrastructure using Infrastructure as Code (IaC) tools and evaluate containerization strategies using Docker and Kubernetes. | PO1(3), PO3(3), PO5(3) | 3 | 3 | 2 |
| **CO4** | Develop and optimize continuous delivery and deployment workflows by integrating monitoring, logging, and feedback mechanisms using tools like Prometheus and ELK. | PO2(3), PO4(2), PO5(2) | 2 | 3 | 3 |
| **CO5** | Assess cloud-native DevOps and integrate security controls into CI/CD pipelines by applying DevSecOps principles and cost-effective cloud deployment strategies. | PO4(3), PO5(3), PO6(3) | 2 | 3 | 3 |

| MC25018 | Ethical Hacking and Penetration Testing | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Prerequisites:**

1. Exposure to computer networks and TCP/IP fundamentals
2. Working knowledge of Linux/Windows OS
3. Familiarity with cyber security fundamentals
4. Programming/scripting knowledge (e.g., Python, Bash) is desirable

**Course Objective**

This course aims to provide a foundational understanding of information security, cyber threats, attacks, and web security. It introduces various hacking tools, penetration testing phases, and methodologies.

**Ethical Hacking Fundamentals**

Ethical Hacking - TCP/IP - IP Addressing - Numbering Systems - Network and Computer Attacks: Malicious Software - Protecting Against Malware Attacks - Intruder Attacks on Networks and Computers - Addressing Physical Security

**Suggested Activities:**

**Activity 1:** Case study to analyze ethical hacking techniques, including vulnerability scanning and network sniffing.

**Activity 2:** Evaluate understanding via quizzes.

**Practical Activities:**

1. Use Wireshark to capture and analyze packets; identify TCP 3-way handshake.
2. Create a Malware Defense Policy for a sample organization.

**Foot Printing and Social Engineering**

Footprinting and Social Engineering: Using Web Tools for Footprinting - Conducting Competitive Intelligence - Using Domain Name System Zone Transfers - Social Engineering - Port Scanning- Port Scanning - Using Port-Scanning Tools - Conducting Ping Sweeps - Understanding Scripting

**Suggested Activities:**

**Activity 2:** Explore and document at least five pre-installed tools used for reconnaissance, vulnerability scanning, exploitation, or social engineering

.

**Practical Activities:**

1. Write a basic shell or Python script to automate Nmap scans and log output.
2. Write a Python script to perform DNS lookups and log the results.

**Penetration Testing And Reconnaissance**

Penetration Testing: Setting the Stage - Kali and Backtrack Linux - Working with Your Attack Machine - The Use and Creation of a Hacking Lab - Phases of A Penetration Test – Reconnaissance - Httrack: Website Copier - Google Directives - The Harvester – Whois – Netcraft – Host - Extracting Information from DNS - Nslookup – Dig – Fierce - Extracting Information from E-Mail Servers – Metagoofil - Threatagent - Social Engineering - Sifting Through the Intel to Find Attackable Targets

**Suggested Activities:**

**Activity 3:** Set up a penetration testing lab using Kali Linux and Backtrack, followed by reconnaissance using tools like Httrack, Whois, and Nslookup.

**Practical Activities:**

1. Set up of Kali Linux in a Virtual machine with attacker and victims
2. Write a Python script to extract hyperlinks from a webpage (web reconnaissance).

**Network Scanning Techniques**

Scanning: Pings And Ping Sweeps - Port Scanning - The Three-Way Handshake - Using Nmap To Perform A TCP Connect Scan - Using Nmap To Perform An Syn Scan - Using Nmap To Perform UDP Scans - Using Nmap To Perform An Xmas Scan - Using Nmap To Perform Null Scans - The Nmap Scripting Engine - Port Scanning Wrap Up - Vulnerability Scanning – Exploitation- Social Engineering

**Suggested Activities:**

**Activity 4:** Perform network scanning using Nmap with various scans to identify open ports and vulnerabilities.

**Activity 5:** Evaluate the effectiveness of scans and exploitation techniques through hands-on lab reports and problem-solving scenarios.

**Practical Activities:**

1. Write a Python script to scan TCP ports and log open ports.

**System Hacking And Password Cracking**

System Hacking Concepts: Understanding Goals of System Hacking – Gaining Access, Escalating Privileges, Maintaining Access, and Clearing Tracks. Password Cracking Techniques: Dictionary Attacks, Brute Force Attacks, Hybrid Attacks. Password Cracking Tools: John the Ripper, Hydra, Cain & Abel, L0phtCrack. Privilege Escalation Techniques. Keylogging and Spyware: Types, Tools, and Countermeasures. Covering Tracks: Clearing Logs, Disabling Auditing, Stealth Techniques. Rootkits and Trojans: Identification, Detection, and Prevention. Real-World Scenarios: Case studies on breaches via weak credentials and poor access controls.

**Suggested Activities:**

**Activity 6:** Evaluate various password-cracking tools in a lab setup and document effectiveness.

**Practical Activities:**

1. Use John the Ripper or Hydra to crack user passwords from a password-protected file or login system.
2. Write a Python script to simulate brute-force attack logic (educational, non-malicious).
3. Simulate a log-clearing operation in Linux and explain its forensic implications.

**Course Outcomes:**

1. Demonstrate knowledge of ethical hacking principles, malware types, and defensive

mechanisms.
2. Apply footprinting, social engineering, and port scanning techniques using common tools.
3. Set up penetration testing labs and perform reconnaissance and exploitation in a simulated environment.
4. Perform network scanning, vulnerability analysis, and reporting using Nmap and scripting techniques.
5. Develop secure policies and defense strategies to protect against real-world cyber threats.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>(i). Activities: 15%<br>(ii). Internal Theory Examinations: 30%<br>(iii) Internal Laboratory Examination: 15% | End Semester Theory Examination: 40% |
| **Mandated Activities with marks:**<br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25). | | |
| **Internal Examinations**: TWO tests | | |

**References**


1. Simpson, M. T., Backman, K., & Corley, J., Hands-On Ethical Hacking and Network Defense, 2nd Edition, Cengage Learning (Course Technology PTR), 2010
2. Engebretson, P., The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy, 2nd Edition, Syngress (Elsevier), August 1, 2013
3. Kim, P., The Hacker Playbook 3: Practical Guide to Penetration Testing, 3rd Edition, Secure Planet, 2025.
4. Weidman, G., “Penetration Testing: A Hands-on Introduction to Hacking”, 2nd Edition, No Starch Press, forthcoming (in progress) — the 2nd edition is currently being worked on, but the 1st Edition (2014).
5. Stallings, W.,” Network Security Essentials: Applications and Standards”, 6th Edition, Pearson, July 2021.
6. Stuttard, D. & Pinto, M., “The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws”, 2nd Edition, Wiley, 2011 (latest available revised edition)


**E–Resources:**

1. OWASP (Open Web Application Security Project)- https://owasp.org
2. Kali Linux Documentation- https://www.kali.org/docs

**CO PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|
| **CO1** | Demonstrate knowledge of ethical hacking principles, malware types, and defensive mechanisms. | PO1(3), PO6(2) | 2 | 2 | - |

| | | | | | |
|---|---|---|---|---|---|
| **CO2** | Apply footprinting, social engineering, and port scanning techniques using common tools. | PO2(3), PO5(2) | 3 | 2 | - |
| **CO3** | Set up penetration testing labs and perform reconnaissance and exploitation in a simulated environment. | PO2(3), PO4(2), PO5(3) | 3 | 3 | 2 |
| **CO4** | Perform network scanning, vulnerability analysis, and reporting using Nmap and scripting techniques. | PO3(3), PO4(3), PO5(3) | 3 | 3 | 2 |
| **CO5** | Develop secure policies and defense strategies to protect against real-world cyber threats. | PO1(3), PO6(3), PO4(2) | 2 | 3 | 3 |

| MC25019 | E-Commerce Technologies | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 2 | 4 |

**Prerequisites:**

1. Basic knowledge of web technologies, including HTML, CSS, and JavaScript.
2. Familiarity with internet and networking fundamentals, such as HTTP, IP, and domain systems.
3. Introductory knowledge of databases and SQL for handling E-commerce data.
4. Basic programming skills, preferably in languages like PHP, Python, or Java.
5. Grasp of business and marketing principles, including online payment systems and customer interaction.
6. Awareness of cybersecurity basics, especially in the context of data privacy and secure transactions.

**Course Objective**:

To provide a comprehensive understanding of E- business systems, covering their evolution, marketing strategies, security mechanisms, and payment solutions. It emphasizes the integration of ERP, CRM, SCM, mobile commerce, and knowledge management to streamline business processes. Students will also explore enabling technologies, real– time E- commerce strategies, and regulatory frameworks like the IT Act 2000.

**Introduction**

History – Early Business Information Interchange Efforts – Emergence of internet– WWW. E- Commerce – Advantages– Disadvantages – BAM Models – Transition to E- Commerce in India– E- Transmission Challenges– The Information Technology Act 2000 – Business Models – Enabling Technologies of WWW.Social Networks– Auctions– Portals: Social networks and online communities – Online auctions – E- commerce portals.

**Suggested Activities:**

**Activity 1:** Timeline Creation – Early Business Information Interchange (EDI)
**Activity 2:** BAM Model Roleplay– Create a mock business using that model.

**Practical Activities:**

1. Create a Timeline of EDI and E-Commerce Evolution
2. Simulate B2B, B2C, and C2C BAM Models through Roleplay

**Electronic Marketing, Security and Payment**

Traditional Marketing – Web Presence Goal – Browsing Behavior Model – Online Marketing– E- Advertising– Trends– E- branding– Strategies. E- Security – Internet Security– E- business Risk Management Issues– Information Security Environment in India. E- Payment – Concerns in Internet Banking– Digital Payment Requirements– Token Based E- Payment Classification– E- Cash– Cheque Payment– Risk and E- Payment.

**Suggested Activities:**

**Activity 3:** Digital Marketing Strategy Simulation
**Activity 4:** Roleplay – Token vs Traditional Payment

**Practical Activities:**
1. Design a Digital Marketing Strategy for an Online Business
2. Compare Token-Based vs Traditional Payment Methods via Simulation

**CRM – SCM & SKM**
E- CRM Solutions – Business Touch Points – Case Studies. Supply Chain – The new way– E- logistics– Fulfilling Customer's Needs– Smart Chains Smarter Gains. Real Time Benefits and Strategies – Advantages. Knowledge as Key Business Asset– Changes in – Global Business Economy– Technology. Knowledge – Definition– Management – Knowledge Management– Data Warehousing and Data Mining. Virtual value chain – 7 Dimensions – E- Commerce Strategy – planning E- Commerce Project.

**Suggested Activities:**

**Activity 5:** Case Study – Customer Touchpoint & E- CRM Strategy Map
**Activity 6:** Hands– on Task – Create a servlet to upload and save files to the server directory.

**Practical Activities:**
1. Develop a Java EE Servlet for File Upload to Server Directory.
2. Map Customer Touchpoints in E-CRM using a Case Study.

**Mobile Commerce and Technologies**
E- Business Portals. What? – Issues – Wireless– Cellular– Wireless Spectrum. – Success Stories. Technologies – mobile commerce- WAP Wireless Generations. Portals – Different Types– benefits– features. Requirements for Intel.

**Suggested Activities:**

**Activity 7:** Presentation – Design Your Own E- Business Portal
**Activity 8:** Wireless Tech Scavenger Hunt + Case Study Relay

**Practical Activities:**
1. Design a Mock UI for a Mobile-Friendly E-Commerce Portal
2. Simulate ERP Process Mapping from Problem to Process Flow

**ERP & EDI**
Introduction – ERP and E2RP – Business Problems– New Paradigm– Drivers – Business processes and supporting processes. Architecture- Implementation– ERP Processes. ERP – Cloud and Open Sources. EDI– Concepts and Technology.

**Suggested Activities:**

**Activity 9:** ERP System Mapping – From Problem to Process
**Activity 10**: EDI & ERP Integration Simulation.

**Practical Activities:**

1. Create a Flow Diagram for EDI and ERP System Integration
2. Conduct a Wireless Technology Scavenger Hunt with Case Study Discussion

**Course Outcomes:**

1. Analyze the evolution of E- Business, early business information exchange efforts, and the emergence of the internet and WWW in business contexts.
2. Explore various E-commerce models, their advantages, disadvantages, and enabling technologies including social networks, portals, and online auctions.
3. Develop knowledge of electronic marketing, web presence strategies, consumer behavior models, E- branding, and online advertising trends.
4. Analyze internet security issues, digital payment systems, E- banking concerns, and risk management in the Indian E- business environment.
5. Examine E- CRM, supply chain management (SCM), knowledge management (SKM), and real– time strategies that enhance E- business performance.

**Assessment Weightage:**

| | | |
|---|---|---|
| **Weightage:** | Continuous Assessment: 60%<br>(i). Activities: 15%<br>(ii). Internal Theory Examinations: 30%<br>(iii) Internal Laboratory Examination: 15% | End Semester Theory Examination: 40% |
| **Mandated Activities with marks:**<br>Assignments (30), Quiz (10), Project based learning (25), Flipped Classroom (10), Review of GATE questions (25). | | |
| **Internal Examinations**: TWO tests | | |

**References:**


1. P. T. Joseph S.J, “E-Commerce – An Indian Perspective”, 7th Edition, PHI Learning Private Limited, 2023
2. Kenneth C. Laudon & Carol Guercio Traver, “E-Commerce: Business – Technology – Society”, 17th Edition, Pearson, 2021
3. Sandeep Desai & Abhishek Srivastava, “ERP to E2RP – A Case Study Approach”, 1st Edition (Eastern Economy), PHI Learning Private Limited, 2013
4. Kamalesh K. Bajaj & Debjani Nag, “E-Commerce – The Cutting Edge of Business”, 2nd Edition, McGraw-Hill Education (India), 2005.
5. Elder Moraes, “Java EE 8 Cookbook: Build reliable applications with the most robust and mature technology for enterprise development”, 1st Edition, Packt Publishing, April 2018.


**E-Resources:**

1. NPTEL Course on E-Business – https://nptel.ac.in/courses/110/105/110105083
2. NPTEL Digital Marketing – https://nptel.ac.in/courses/110/105/110105150
3. Java EE Tutorials – https://youtube.com/@JavaBrainsChannel
4. W3Schools Java Servlet Guide – https://www.w3schools.com/java/java_servlets.asp

5. SWAYAM Portal – https://swayam.gov.in

**CO-PO Mapping:**

| CO No. | Course Outcome Description | PO Mapping | PSO 1 | PSO 2 | PSO 3 |
|---|---|---|---|---|---|
| **CO1** | Analyze the evolution of E-Business, early business information exchange efforts, and the emergence of the internet and WWW in business contexts. | PO1(3), PO6(2) | 2 | 1 | - |
| **CO2** | Explore various E-commerce models, their advantages, disadvantages, and enabling technologies including social networks, portals, and online auctions. | PO2(3), PO3(2) | 2 | 2 | - |
| **CO3** | Develop knowledge of electronic marketing, web presence strategies, consumer behavior models, E-branding, and online advertising trends. | PO2(2), PO4(3), PO5(2) | 2 | 2 | 1 |
| **CO4** | Analyze internet security issues, digital payment systems, E-banking concerns, and risk management in the Indian E-business environment. | PO3(3), PO4(2), PO6(3) | 2 | 3 | 2 |
| **CO5** | Examine E-CRM, supply chain management (SCM), knowledge management (SKM), and real–time strategies that enhance E-business performance. | PO1(2), PO5(3), PO6(2) | 3 | 3 | 2 |

| MC25020 | Social Network Analysis | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |

**Course Objective:**

1. To understand the basics of social networks and its modelling.
2. To apply and hone their data analysis skills on social media data to find meaningful patterns
3. To apply techniques for link prediction, community detection and graph embedding in social networks
4. To carry out some case studies in social network analysis.

**Introduction to Social Network Analysis**

Social Network - Types of social networks - Measurement and collection of social network data - Opportunities and challenges in social networks Analysis - Real-world use cases of Social Network Analysis - Graphs: Networks and representations – Nodes – Edges – Degree - Directed and Undirected graphs- Weighted and Unweighted graphs- Representations - Adjacency matrices - lists – Paths – Components – Connectivity - Cliques - Cut sets.

**Network Measures and Models:**

Network Measures: Centrality - Degree centrality - Betweenness centrality - Closeness centrality - PageRank - Hubs and Authorities - Transitivity and Reciprocity - Similarity - Network Growth Models: Random Network Model; Ring Lattice Network Model; Watts–Strogatz Model; Preferential Attachment Model; Price's Model; Local-world Network Growth Model.

**Community Structure in Networks:**

Applications of Community Detection - Types of Communities - Community Detection Methods - modularity maximization, label propagation, Louvain - Disjoint Community Detection - Overlapping Community Detection - Local Community Detection - Community Detection vs Community Search - Evaluation of Community Detection Methods

**Link Prediction**

Applications of Link Prediction, Temporal Changes in a Network; Evaluating Link Prediction Methods; Heuristic Models; Probabilistic Models; Supervised Random Walk; Information-theoretic Model; Latest Trends in Link Prediction.

**Graph Representation Learning**

Machine Learning Pipelines; Intuition behind Representation Learning; Benefits of Representation Learning; Criterion for Graph Representation Learning; Graph Representation Learning Pipeline; Representation Learning Methods - Models: Node2Vec, GraphSAGE, GCNs

**Real-world use cases of Social Network Analysis:**

Sentiment Analysis on a Movie Reviews dataset - Recommendation Systems - Empirical Study Social media analysis (Twitter, Facebook) - Political networks Collaboration networks (e.g., co-authorship)

**Course Outcomes:**

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

- **CO1:** Understand the concept and structure of social networks and its implications.
- **CO2:** Enable students to practically analyze social network data and how to reason about it through models for network structure and evolution.
- **CO3:** Apply techniques for link prediction, community detection and graph embedding in social networks
- **CO4:** Apply social network analysis to real-world problems, such as recommendation systems, social media analytics, sentiment analysis, and political or collaborative networks.

| CO | PO Mapping | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|
| CO1 | PO1 (3)<br>PO2 (3)<br>PO3 (2) | 2 | - | - |
| CO2 | PO2 (2)<br>PO3 (3)<br>PO4 (2)<br>PO5 (1)<br>PO10 (2) | 1 | - | 3 |
| CO3 | PO3 (3)<br>PO4 (2)<br>PO5 (1) | 3 | 1 | - |
| CO 4 | PO1 (1)<br>PO8 (2)<br>PO10 (3)<br>PO11 (2) | 1 | 2 | 3 |

**Assessment Weightage:**

**Continuous assessment weightage: 40%**

**End semester examination weightage: 60%.**

Continuous assessments comprise of two assessments and 2 activities.

- Assessment 1: 40%
- Assessment 2: 40%
- Activity: 20%

**# Activities:**

1. Flipped Classroom: Analyze and critique different community detection algorithms.
2. Capstone Project: Analyze Twitter data for sentiment or political influence; Build a recommendation engine using graph-based features.

**References:**


1. Tanmoy Chakraborty," Social Network Analysis", Wiley, 2021.
2. Matthew A. Russell, Mikhail Klassen, "Mining the Social Web", 3rd Edition, O'Reilly, 2019.
3. Albert-Lazzlo Barabasi, "Network Science", Cambridge University Press, 2016.
4. Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. Social media mining: an introduction. Cambridge University Press, 2014
5. Stanley Wasserman, Katherine Faust, "Social Network Analysis: Methods and Applications", Cambridge University Press, 2012
6. William L. Hamilton, Graph Representation Learning, Morgan & Claypool Publishers, 2020
7. NPTEL Course: https://nptel.ac.in/courses/106106239