# DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

## VISION:

To be recognized as a center for quality education in the field of mechanical engineering, integrating top-tier teaching, innovative research, and fostering socially conscious and entrepreneurial engineers.

## MISSION:

**M1:** To impart value-added education by creating an ambiance of academic excellence in teaching-learning processes.
**M2:** To inculcate a research approach through innovation and skill development centers.
**M3:** To inculcate a strong sense of social responsibility and empathy among the students.
**M4:** To imbibe the entrepreneur skill amongst the students by strengthening Industry- Institute interaction.

## PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

PEO1: The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, networking, artificial intelligence and data science for efficient design of computer-based systems of varying complexities.
PEO2: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PEO3: Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

## PROGRAM OUTCOMES (POs):

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2: **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with

appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSOs):**

PSO1: The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, networking, artificial intelligence and data science for efficient design of computer-based systems of varying complexities.

PSO2: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3: Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

# LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| BSC | Basic Science Course |
| ESC | Engineering Science Course |
| PCC | Programme Core Course |
| PEC | Programme Elective Course |
| MDM | Multidisciplinary Minor |
| OE | Open Elective - Other than a particular program |
| VSEC | Vocational and Skill Enhancement Course |
| AEC | Ability Enhancement Course |
| ENTR | Entrepreneurship |
| EC | Economics |
| MC | Management Courses |
| IKS | Indian Knowledge System |
| VEC | Value Education Courses |
| RM | Research Methodology |
| CEP | Community  Engagement Project |
| FP | Field Project |
| PROJ | Project |
| INT | Internship |
| OJT | On Job Training |
| CC | Co-curricular Courses |
| HSSM | Humanities Social Science and Management |
| ELC | Experiential Learning Course |
| B. Tech | Bachelor of Technology |
| L | Lecture |
| P | Practical |
| T | Tutorial |
| H | Hours |
| CR | Credits |
| IE | Internal Evaluation |
| MTE | Mid Term Evaluation |
| ETE | End Term Evaluation |
| TW | Term Work |
| OR | Oral |
| PR | Project |

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

### Second Year B. Tech. – AI & ML: Semester - III

| Course Code | Course Type | Course Name | Teaching Scheme (hrs/Week) | | | | | | | Evaluation Scheme | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | P | T | H | CR | | | CIE | ETE | TW | PR | OR | Total |
| | | | | | | | TH | PR/ Tut | Total | | | | | | |
| AMPC302 | PCC | Data Structures | 3 | 4 | - | 7 | 3 | 2 | 5 | 40 | 60 | - | 25 | - | 125 |
| AMPC303 | PCC | Object Oriented programming | 2 | - | | 2 | 2 | - | 2 | 40 | 60 | - | - | - | 100 |
| AMPC304 | PCC | Operating Systems | 2 | 2 | - | 4 | 2 | 1 | 3 | 40 | 60 | - | - | 25 | 125 |
| AMMD301 | MDM | Statistics for Artificial Intelligence | 3 | - | - | 3 | 3 | - | 3 | 40 | 60 | - | - | - | 100 |
| ALOE301 | OE | Open Elective-I | 2 | - | - | 2 | 2 | - | 2 | 40 | 60 | - | - | - | 100 |
| AMMC301 | HSSM-MC | Management Information System | 1 | - | 1 | 2 | 1 | 1 | 2 | - | - | 25 | - | - | 25 |
| AMVS303 | VSEC | Programming Lab I | - | 4 | - | 4 | - | 2 | 2 | - | - | 25 | 25 | - | 50 |
| AMCE301 | CEP | Mini Project | - | 2 | - | 2 | | 1 | 1 | - | - | 50 | - | - | 50 |
| AMIN302 | ELC - INT | Internship-II | 4 Weeks | | | | - | 2 | 2 | - | - | 25 | - | - | 25 |
| **Total** | | | **13** | **12** | **-** | **26** | **13** | **9** | **22** | **200** | **300** | **125** | **50** | **25** | **700** |

| # - Select any one course from the given Open Elective Courses | | |
|---|---|---|
| **Course Code** | **Course Type** | **Open Elective - I** |
| ALOE301A | OEC | Digital Literacy and Applications |
| ALOE301B | | Environmental Studies |
| ALOE301C | | Green Energy and Sustainability |
| ALOE301D | | Basics of Consumer Electronics |
| ALOE301E | | Renewable Energy Systems |

BoS Chairman

Director

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

### Second Year B. Tech. – AI & ML: Semester - IV

| Course Code | Course Type | Course Name | Teaching Scheme | | | | | | | Evaluation Scheme | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | CR | | | | | | | | |
| | | | L | P | T | H | TH | PR/Tut | Total | CIE | ETE | TW | PR | OR | Total |
| AMPC405 | PCC | Database Management System | 3 | 2 | - | 5 | 3 | 1 | 4 | 40 | 60 | - | **50** | - | 150 |
| AMPC406 | PCC | Data Science | 2 | 2 | - | 4 | 2 | 1 | 3 | 40 | 60 | - | - | 25 | 125 |
| AMPC407 | PCC | Software Engineering | 3 | - | - | 3 | 2 | 1 | 3 | 40 | 60 | - | - | - | 100 |
| AMMD402 | MDM | Knowledge Representation and Reasoning | 3 | - | - | 3 | 3 | - | 3 | 40 | 60 | - | - | - | 100 |
| ALOE402 | OE | Open Elective - II # | 2 | - | - | 2 | 2 | - | 2 | 40 | 60 | - | - | - | 100 |
| AMMC402 | HSSM-MC | E-Commerce | 1 | - | 1 | 2 | 1 | 1 | 2 | - | - | 25 | - | - | 25 |
| AMAE402 | AEC | Quantitative Reasoning and Analysis | - | 2 | - | 2 | - | 1 | 1 | - | - | 25 | - | - | 25 |
| AMVS404 | VSEC | Programming Lab II | - | 4 | - | 4 | - | 2 | 2 | - | - | 25 | 25 | - | 50 |
| AMIN403 | ELC - INT | Internship-III | 4 Weeks | | | | - | - | 2 | - | - | 25 | - | - | 25 |
| **Total** | | | **14** | **10** | **1** | **25** | **13** | **11** | **22** | **200** | **300** | **100** | **75** | **25** | **700** |

| # - Select any one course from the given Open Elective Courses | | |
|---|---|---|
| **Course Code** | **Course Type** | **Open Elective - II** |
| ALOE402A | OEC | Cyber Security and Laws |
| ALOE402B | | Sustainability and Climate Change |
| ALOE402C | | Energy Audit and Electrical Safety |
| ALOE402D | | Digital Marketing |
| ALOE402E | | Entrepreneurship and Innovations |

BoS Chairman

Director

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

# INDEX

# Semester-III

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** III | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Data Structures | | | | | | | | **Code:** AMPC302 | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 03 | 04 | - | 07 | 40 | 60 | - | - | 25 | 125 |

**Prerequisites:**

- Basic knowledge of programming (preferably in C/C++)
- Understanding of control structures and functions

**Course Objectives:**

- To understand and implement fundamental data structures and algorithms.
- To develop problem-solving skills and algorithmic thinking.
- To analyze time and space complexity for efficient code design.
- To apply appropriate data structures for real-life applications.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Understand the basic concepts of data structures and algorithm design. |
| **CO2** | Implement linear and nonlinear data structures using programming techniques. |
| **CO3** | Apply stack, queue, linked list, and tree structures in various applications. |
| **CO4** | Analyze and compare various searching and sorting algorithms. |
| **CO5** | Design efficient algorithms using appropriate data structures. |
| **CO6** | Evaluate the performance and complexity of algorithms. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Data Structures and Algorithms:** Need and importance of data structures in problem-solving. Characteristics of a good algorithm. Difference between data types and data structures. **Classification of data structures** – linear and non-linear. Static vs dynamic data structures. **Overview of Abstract Data Types (ADT)** – List, Stack, Queue. Comparison between recursive and iterative approaches. Common issues faced in programming without efficient data structures. | 7 Hrs |
| 2. | **Arrays and Linked Lists:** **Concept and representation of arrays, operations** – insertion, deletion, traversal, and searching. Static vs dynamic memory allocation. | 6 Hrs |

| | | |
|---|---|---|
| | **Linked Lists** – Singly, Doubly, and Circular. Implementation of linked lists using pointers. Applications of linked lists – polynomial addition, dynamic memory usage. | |
| 3. | **Stacks and Queues**:<br>Stack definition and operations – push, pop, peek. Implementation using arrays and linked lists. Applications – expression conversion, expression evaluation, and recursion handling. Queue definition and types – Simple Queue, Circular Queue, Priority Queue, Double-Ended Queue (Deque). Queue operations and their real-life applications like CPU scheduling and buffering. | 5 Hrs |
| 4. | **Trees:**<br>**Basic terminology and representation of trees. Binary Tree** – creation and traversals (inorder, preorder, postorder). Binary Search Tree – insertion, deletion, and search. **Introduction to AVL Trees** – need and rotations. Heap – Min Heap, Max Heap and Heap Sort. Applications of trees – expression trees, Huffman coding. | 6 Hrs |
| 5. | **Graphs:**<br>**Introduction to graphs** – definitions and terminologies. Representations – adjacency matrix and adjacency list. Graph traversal techniques – BFS and DFS. **Minimum Spanning Tree** – Prim's and Kruskal's algorithms. Dijkstra's algorithm for shortest path. Applications of graphs – routing, social networks, topological sorting. | 5 Hrs |
| 6. | **Searching and Sorting Algorithms:**<br>**Searching techniques** – Linear and Binary Search (recursive and iterative).<br>**Sorting techniques** – Bubble, Selection, Insertion, Merge, Quick, and Heap Sort. Comparison of sorting algorithms based on time and space complexity. Real-world usage of searching and sorting in databases and data processing. | 7 Hrs |
| | **TOTAL** | 36 Hrs |

**Text Books:**

1. Ellis Horowitz, Sartaj Sahni, "Fundamentals of Data Structures", Galgotia Publications.
2. Reema Thareja, "Data Structures using C", Oxford University Press.

**Reference Books:**

- Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education.
- Seymour Lipschutz, "Data Structures", Schaum's Outline Series.

**E-Resources:**

- NPTEL Online Courses: https://nptel.ac.in
- GeeksforGeeks: https://www.geeksforgeeks.org
- Coursera – Data Structures: https://www.coursera.org

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | **Semester:** III |
|---|---|
| **Course:** Data Structures and Algorithms Lab | **Code:** AMPC303 |

**Guidelines for Instructor's Manual**

The instructor's manual should briefly cover the university/program background, course objectives, syllabus, lab conduction methods, and assessment criteria. It should provide topic-wise concepts, expected outcomes, and a list of practical assignments using languages such as **C, C++, or Java**. Sample code structures, algorithmic logic, and references should be included to support effective lab delivery. Standard evaluation rubrics, viva questions, and clarity on CO (Course Outcome) mapping must be provided. The manual is intended to ensure consistency in lab sessions and support instructors in aligning activities with the desired learning outcomes.

**Guidelines for Student's Lab Journal**

The student's lab journal should follow a uniform structure containing the experiment title, objective, problem statement, algorithm or pseudocode, program code, sample input/output, and conclusion. It should reflect individual work and understanding. Each journal must include a content index and a summary of weekly progress. Code documentation and neat presentation are expected. This record serves as the student's learning evidence and must be duly signed and verified each week by the lab instructor.

**Guidelines for Lab /TW Assessment**

The term work should be continuously assessed throughout the semester. Marks should be allotted based on timely submissions, code functionality, understanding of concepts, journal completion, and viva performance. Experiments should be graded using standardized rubrics focusing on logic design, correctness, output, and documentation. Bonus weightage may be given for case study analysis and the mini-project. The goal is to promote regular engagement, practical proficiency, and overall conceptual clarity.

**Guidelines for Practical Examination**

The practical examination should assess the student's coding ability, problem-solving skills, and conceptual understanding. It should consist of one programming problem from the performed experiments and an oral viva. Students must write and execute the solution independently within the given time frame. Evaluation should be based on execution correctness, code structure, and oral response, using standard rubrics. Both internal and external examiners must align on evaluation methods.

**Guidelines for Laboratory Conduction**

Laboratory sessions should begin with a brief explanation of the topic, objectives, and expected outcomes. Instructors should encourage active participation, individual coding practice, and peer learning. Faculty should monitor progress during the session and resolve student doubts. Weekly viva or discussion should be conducted for conceptual reinforcement. A lab timetable, attendance record, and

| |
|---|
| assessment log should be maintained. The aim is to build consistent technical skills and analytical thinking through hands-on practice. |
| **List of Assignments:- (Perform Any 10 Practical's)** |
| 1. **Array Operations — CO1** |
| Implement insertion, deletion, searching, and traversal on one-dimensional and two-dimensional arrays using C, C++, or Java. Accept dynamic input and provide user-friendly menus. Analyze time complexity for each operation. |
| 2. **Linked List Implementation — CO1, CO2** |
| Develop programs for singly, doubly, and circular linked lists. Perform insertion and deletion at various positions. Display elements and handle edge cases like underflow and overflow. |
| 3. **Stack Implementation and Applications — CO2, CO3** |
| Implement stack using arrays and linked lists. Perform infix to postfix conversion and evaluate postfix expressions. Use stack to check balanced parentheses in expressions. |
| 4. **Queue Variants — CO2, CO3** |
| Implement simple queue, circular queue, and priority queue using arrays. Include operations for enqueue, dequeue, peek, and display. Simulate a basic scheduling system using queue. |
| 5. **Binary Search Tree (BST) — CO3, CO4** |
| Construct a BST and perform insertion, deletion, and traversal (inorder, preorder, postorder). Display the tree in a structured format. Highlight applications in searching and sorting systems. |
| 6. **Heap and Heap Sort — CO3, CO4** |
| Implement max heap and min heap. Perform heap sort using heapify and demonstrate sorting steps. Explain the role of heaps in priority-based task handling. |
| 7. **Graph Traversal — CO4** |
| Use adjacency matrix or list to represent graphs. Implement BFS and DFS traversal. Provide input for directed and undirected graphs and display the visited sequence clearly. |
| 8. **Shortest Path Algorithm — CO4** |
| Implement Dijkstra's algorithm to find the shortest path in a weighted graph. Accept graph input from the user and display step-wise output with a distance table. |
| 9. **Sorting Algorithms Comparison — CO4, CO5** |
| Implement and compare sorting algorithms: Bubble, Insertion, Merge, Quick, and Heap sort. Analyze and tabulate their performance based on time and number of comparisons for various inputs. |
| 10. **Searching Techniques — CO4, CO5** |
| Implement linear and binary search algorithms (iterative and recursive). Perform complexity analysis. Accept unsorted and sorted input lists and demonstrate differences in performance. |

| 11. **Case Study: Application of Data Structures — CO6** |
|---|
| **Choose any one case:**<br>a) Railway reservation system using linked list and queue<br>b) Syntax checker using stack<br>c) Graph-based social network connection map<br>Apply one or more data structures to build a simplified prototype and document usage with code and explanation. |
| 12. **Mini Project — CO6** |
| Design and develop a mini project using multiple data structures. Suggested ideas:<br>● Student record manager using files and trees<br>● Command-line text editor with undo feature (stack)<br>● File directory simulation using tree structure<br>Submit code, execution video/screenshots, and a short project report including problem definition, system design, and implementation details. |
| **Text Books:** |
| 1. Ellis Horowitz, Sartaj Sahni, "Fundamentals of Data Structures", Galgotia Publications.<br>2. Reema Thareja, "Data Structures using C", Oxford University Press. |
| **Reference Books:** |
| ● Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education.<br>● Seymour Lipschutz, "Data Structures", Schaum's Outline Series. |
| **E-Resources:** |
| ● NPTEL Online Courses: https://nptel.ac.in<br>● GeeksforGeeks: https://www.geeksforgeeks.org<br>● Coursera – Data Structures: https://www.coursera.org |

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** III | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Object Oriented Programming | | | | | | | | **Code: AMPC303** | |
| **Teaching Scheme (Hrs/week)** | | | | **Evaluation Scheme (Marks)** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 02 | - | - | 02 | 40 | 60 | - | - | - | 100 |

**Prerequisites:**

Basic Programming Concepts, Procedural Programming Knowledge, Pointers and Memory Management, Mathematical Foundations, Basic Understanding of Algorithms

**Course Objectives:**

1. To introduce the fundamental programming paradigms and demonstrate the shift from procedural to object-oriented programming.
2. To explore key OOP principles such as classes, inheritance, polymorphism, and encapsulation.
3. To provide hands-on experience in solving problems using C++.
4. To enable students to handle advanced features like operator overloading, file handling, templates, and exception handling.
5. To strengthen the ability to design and implement robust programs following object-oriented methodologies.

**Course Outcomes:**

| | |
|---|---|
| **CO1** | Students will be able to differentiate between various programming paradigms and apply basic object-oriented concepts in C++ |
| **CO2** | Students will understand and implement different types of inheritance using C++ |
| **CO3** | Students will be able to implement static and dynamic polymorphism in C++ programs |
| **CO4** | Students will develop generic programs using templates and manage exceptions in C++ programs. |
| **CO5** | Students will implement file handling techniques and manipulate file data using C++ |
| **CO6** | Students will handle advanced OOP concepts such as dynamic memory management and STL usage. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Programming Paradigms:**<br>Role and importance of programming languages, Characteristics of good programming languages<br>**Overview of programming paradigms**: Procedural, Object-Oriented, Logical, and Functional,<br>Comparison between Procedural and Object-Oriented Programming (OOP).<br>**Features of Object-Oriented Programming (OOP):** Abstraction, Encapsulation, Inheritance And Polymorphism.<br>**C++ Syntax**: Data types, Variables, Operators, Flow Control, Arrays, Pointers<br>**C++ Classes**: Private, Public, Constructors, Destructors, Member Data, Member Functions | 7 |

| | | |
|---|---|---|
| 2. | **Inheritance**:<br>Concept of Class Hierarchy and Derived Classes,<br>**Types of Inheritance**: Single, Multiple, Multilevel, and Hybrid Inheritance<br>Role of Virtual Base Class, Constructor and Destructor Execution in Derived Classes ,Base Class Initialization using Derived Class Constructors | 7 |
| 3. | **Polymorphism**:<br>Static Binding and Dynamic Binding, Static Polymorphism: Function Overloading, Ambiguity in Function Overloading, Operator Overloading (Unary and Binary Operators), Operator Overloading Using Friend Functions, String Manipulation using Operators, **Dynamic Polymorphism:** Base Class Pointers, Object Slicing, Method Overriding, Virtual Functions, Pure Virtual Functions, Abstract Classes | 7 |
| 4. | **Generic Programming & Exception Handling**:<br>Introduction to Generic Programming, Function Templates, Class Templates, Templates with Multiple Parameters, Exception Handling: Fundamentals, Multiple Catch Blocks, Nested try Statements, Uncaught Exceptions, Stack Unwinding, throw and rethrow | 7 |
| 5. | **File Handling in C++:**<br>Introduction to File Handling and File Stream Classes, File Operations: Opening, Closing Files, Detecting End of File (EOF), File Modes, File Pointer Manipulation and Sequential I/O Operations, Random Access to Files, Error Handling during File Operations | 7 |
| 6. | **Advanced Object-Oriented Concepts:**<br>Operator Overloading: Friend Functions, Assignment Operators<br>Dynamic Memory Management: new and delete operators, Smart Pointers and Reference Counting, Introduction to Standard Template Library (STL): Containers, Iterators, Algorithms, Real-world Case Studies in Object-Oriented Design. | 7 |
| | **TOTAL** | 42 |

**Text Books:**

1. E. Balagurusamy, Object-Oriented Programming with C++, McGraw Hill.
2. Bjarne Stroustrup, The C++ Programming Language, Addison-Wesley
3. Robert Lafore, Object-Oriented Programming in *C++*, SAMS Publishing

**Reference Books:**

1. Herbert Schildt, *C++:* The Complete Reference, McGraw Hill
2. Stanley B. Lippman, C++ Primer, Addison-Wesley
3. Scott Meyers, Effective C++, Addison-Wesley

**E-Resources:**

1. NPTEL: https://onlinecourses.nptel.ac.in/noc21_cs02/preview
2. W3Schools: https://www.w3schools.com/cpp/
3. Coursera: https://www.coursera.org/learn/object-oriented-cpp?specialization=hands-on-cpp
4. GeeksforGeeks: https://www.geeksforgeeks.org/c-plus-plus/

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | **Semester:** III | | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:**Operating Systems | | | | | | | **Code: AMPC304** | | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 02 | 01 | - | 03 | 40 | 60 | - | 25 | - | 125 |

**Prerequisites:**

- Fundamentals of Computer Systems
- Basic knowledge of C programming
- Familiarity with command-line interfaces (optional but desirable)

**Course Objectives:**

- To understand the fundamental concepts and design principles of operating systems.
- To apply OS concepts practically through the Linux operating system.
- To develop skills in Linux commands, shell scripting, and process handling.
- To explore memory management, file systems, and system calls using Linux.
- To gain hands-on experience with real-world Linux environments.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Describe the structure and functionalities of modern operating systems. |
| **CO2** | Use Linux commands and shell scripts for performing system-level tasks. |
| **CO3** | Apply concepts of process management and inter-process communication in Linux. |
| **CO4** | Analyze and apply memory and file management strategies using Linux tools. |
| **CO5** | Develop simple system-level programs using Linux APIs and system calls. |
| **CO6** | Demonstrate practical skills in Linux system administration and automation. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **UNIT 1: Introduction to Operating Systems and Linux**<br>OS types and structures, Functions of an Operating System, Overview of Linux and its architecture, Linux installation, booting process, Basic Linux shell and command-line interface | 7 Hrs |
| 2. | **UNIT 2: Linux Commands and Shell Scripting**<br>File handling, process commands, filters, Bash shell scripting: variables, loops, functions, Cron jobs, script automation, Text processing with awk, sed, grep | 7 Hrs |
| 3. | **UNIT 3: Process and Thread Management**<br>Process states and lifecycle**,** Process creation: fork(), exec()**,** Multithreading basics: pthreads**,** Signals and process synchronization | 7 Hrs |

| | | |
|---|---|---|
| 4. | **UNIT 4: Memory Management in Linux**<br>Address binding, paging, segmentation**,** Virtual memory and demand paging**,** Shared memory, mmap(), memory allocation**,** Linux memory information tools | 7 Hrs |
| 5. | **UNIT 5: File Systems and I/O Management**<br>File structure and hierarchy in Linux**,** System calls: open(), read(), write(), close()**,** File permissions, links (soft & hard), inodes**,** Mounting and file system administration, | 7 Hrs |
| 6. | **UNIT 6: Linux System Administration and Case Studies**<br>User & group management, file permissions**,** Disk partitioning, LVM basics**,** Package management (RPM, apt)**,** OS security basics and log analysis**.**<br>Case Study: Mini-project on shell scripts for admin tasks | 7 Hrs |
| | **TOTAL** | 42 Hrs |

**Text Books:**

1. Operating System Concepts – Silberschatz, Galvin, Gagne (Wiley)
2. Linux Command Line and Shell Scripting Bible – Richard Blum & Christine Bresnahan (Wiley)

**Reference Books:**

- Modern Operating Systems – Andrew Tanenbaum (Pearson)
- Understanding the Linux Kernel – Daniel Bovet, Marco Cesati (O'Reilly)
- Advanced Programming in the UNIX Environment – W. Richard Stevens (Addison-Wesley)

**Digital References:**

- https://linuxjourney.com – Free Linux learning platform
- https://explainshell.com – Shell command explanations
- https://man7.org – Linux manual and system programming guides
- YouTube: "Linux Essentials" by The Linux Foundation
- NPTEL Course: "Operating System Fundamentals" – by Prof. P. K. Biswas (IIT Kharagpur)

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| | |
|---|---|
| **Program:** B. Tech. (AI & ML ) | **Semester:** III |
| **Course:** Operating Systems | **Code: AMPC304** |

**Guidelines for Instructor's Manual**

The instructor's manual should include background, course objectives, CO mapping, unit-wise lab goals, practical outcomes, sample code (in C/shell), screenshots, and rubrics for evaluation. It should facilitate consistent conduction and effective outcome-based assessment.

**Guidelines for Student's Lab Journal**

The journal must contain a certificate, index, aim, objectives, algorithm/code, execution steps, output, conclusion, and teacher's sign for each lab. Students should also attach terminal output screenshots or analysis observations as proof of execution.

**Guidelines for Lab /TW Assessment**

Assessment is continuous based on journal submission, code quality, viva performance, lab attendance, and timely completion. Mini projects and additional work (e.g., automation scripts) may receive bonus points. Rubrics should reflect CO attainment.

**Guidelines for Practical Examination**

Each student will be given a task from the lab syllabus involving Linux commands, system calls, or shell scripts. Evaluation will include code correctness, clarity of logic, documentation, and viva. Practical work must demonstrate applied OS knowledge.

**Guidelines for Laboratory Conduction**

Start with a short briefing of theory and Linux concepts. Demonstrate related commands and scripting live. Students should work hands-on in a Linux terminal or emulator (VM or dual boot). Encourage command-line practice and Linux man pages.

**List of Assignments:- (Perform Any 10 Practical's)**

1. **Assignment :- 01**

Execute and demonstrate at least 20 basic Linux commands related to file handling, directory navigation, and user interaction. Include commands like ls, pwd, cd, mkdir, rm, cat, touch, chmod, chown, whoami, passwd, df, and du.

2. **Assignment :- 02**

Explore and analyze the Linux file system hierarchy. Use ls -l, chmod, and stat to inspect and modify file permissions. Demonstrate effects for user, group, and others with examples.

3. **Assignment :- 03**

Write shell scripts to:
– Calculate the factorial of a number using loops
– Display system information such as hostname, uptime, logged-in users, and disk usage

4. **Assignment :- 04**

Create a shell script to back up a directory into a .tar.gz file. Schedule the task using cron jobs. Validate scheduled execution with timestamps or logs.

| |
|---|
| 5. **Assignment :- 05** |
| Use awk, sed, and grep for text processing:<br>– Extract valid email IDs from a file<br>– Replace specific text<br>– Filter logs with keywords like "error" or "warning" |
| 6. **Assignment :- 06** |
| Write a C program using fork(), exec(), and wait() to create a child process that runs a different program. Display parent and child process information. |
| 7. **Assignment :- 07** |
| Develop a multithreaded C program using pthread to simulate a producer-consumer problem with shared buffers. Use mutex locks and condition variables. |
| 8. **Assignment :- 08** |
| Write a C program to set up a custom signal handler for SIGINT or SIGTERM. On signal receipt, display a custom message and exit gracefully. |
| 9. **Assignment :- 09** |
| Write a program using shared memory (shmget(), shmat(), shmdt()) or mmap() to allow data exchange between processes. Demonstrate both read and write. |
| 10. **Assignment :- 10** |
| Analyze memory usage using /proc/meminfo, top, free, and vmstat. Capture observations under different system conditions. |
| 11. **Assignment :- 11** |
| Write a C program to copy file contents using system calls (open(), read(), write(), close()). Also show how to modify file permissions and create symbolic and hard links. |
| 12. **Assignment :- 12** |
| Write automation shell scripts for basic system admin tasks:<br>– Create multiple users and groups<br>– Install packages using apt or rpm<br>– Monitor disk usage and services using df, systemctl, journalctl |

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (Artificial Intelligence And Data Science) | | | | | | | | | **Semester:** III |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Statistics for AI | | | | | | | | | **Code: AMMD301** |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 3 | - | - | 03 | 40 | 60 | - | - | - | 100 |

**Prerequisites:**

Knowledge of probability and Statistics

**Course Objectives:**

To enable students to understand and apply statistical concepts and methodologies including bivariate distributions, hypothesis testing, regression analysis, time series modeling, and real-time AI applications, thereby preparing them to analyze and interpret data for AI-based decision-making and systems development.

**Course Outcomes: After completion of this course, student will be able to -**

| | |
|---|---|
| **CO1** | Understand and apply the concepts of discrete and continuous random variables, including joint, marginal, and conditional probability distributions for bivariate cases. |
| **CO2** | Demonstrate the ability to construct and interpret statistical hypotheses, including null and alternative hypotheses, and evaluate their significance using statistical inference principles. |
| **CO3** | Apply various hypothesis testing techniques, including likelihood ratio tests and non-parametric methods, to draw meaningful conclusions from sample data. |
| **CO4** | Analyze relationships between variables using correlation and regression methods, and apply these techniques in AI applications such as feature selection and prediction models.. |
| **CO5** | Develop and evaluate time series models such as ARIMA, SARIMA, and exponential smoothing for trend analysis and forecasting in real-world datasets. |
| **CO6** | Identify and explain the role of statistical models in real-time AI applications including autonomous systems, recommendation engines, and fraud detection systems. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|

| | | |
|---|---|---|
| 1. | **Random Variables & Bivariate Probability Distribution**<br>Introduction,Random Variable : discrete,continuous with example<br>Bivariate random variable: discrete, continuous, Bivariate probability,distribution,Marginal,Probability Distribution,Independence of Two Discrete Random Variable,Conditional Probability Distribution<br>Joint Distribution Function of Two Dimensional Discrete R.V. | 7Hrs |
| 2. | **Inferential Statistics: Hypothesis**<br>Statistical Inference - Testing of Hypothesis, Non-parametric Methods and Sequential,Analysis:Introduction,Statistical Hypothesis (Simple and-Composite), Test of a Statistical Hypothesis,Null Hypothesis, Alternative Hypothesis, Critical Region, level of Significance, Power of the Test, Principle of likelihood. | 7Hrs |
| 3. | **Inferential Statistics: Tests For Hypothesis**<br>Most Powerful Test (MP Test), Uniformly Most Powerful Test, likelihood Ratio Test,Properties of Likelihood Ratio Test, Test for the Mean of a Normal Population, Test for the Equality of Means of Two Normal Populations, Test for the Equality of -Means of Several Normal Populations, Test for the Variance of a Normal Population, Test for Equality of Variances of two Normal Populations, Non-parametric Methods, Advantages and<br>Disadvantages of Non-parametric Methods. | 7Hrs |
| 4. | **Regression & Correlation**<br>Concept of correlation, Types: Positive, Negative, Zero, AI Application: Correlation in feature selection<br>Regression, types of regression (Simple linear regression, Logistic Regression,, Multiple Linear Regression) | 7Hrs |
| 5. | **Time Series Analysis**<br>Introduction, Components of time series data MA model – basic and weighted MA model Time series models AR Model, ARIMA Model, SARIMA,SARIMAX,VAR,VARMAX, Simple exponential smoothing model. | 7Hrs |

| | | |
|---|---|---|
| 6. | **Real-Time Applications in AI**<br>1.Autonomous Vehicles<br>2.Chatbots and Virtual Assistants<br>3.Facial Recognition Systems<br>4.Real-Time Fraud Detection<br>5.Personalized Recommendations<br>6.Industrial Robotics and Automation | 7 Hrs |
| | | **42Hrs** |

**Text Books:**

1. Statistics (Dr.P.G.Dixit)
1. Statistics for Data Scientsists, An introduction to probability, statistics and Data Analysis, MauritsKaptein et al,Springer 2022
2. Probability and Statistics for Engineering and Sciences,8th Edition, Jay L Devore, Cengage Learning
3. Introduction to Time Series and Forecasting, Second Edition, Peter J Brockwell, Richard A Davis, Springer.

**E-Resources:**

**MOOC / NPTEL/YouTube Links:**

1. https://www.w3schools.com/html/html_blocks.asp

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | | **Semester:** III |
|---|---|---|---|---|---|---|---|---|---|
| **CCourse: Open Elective-I [Digital Literacy and Applications]** | | | | | | | | | **Code: ALOE301** |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 02 | - | - | 02 | 40 | 60 | - | - | - | 100 |

**Prerequisites:**

- Basic computer literacy and fundamental programming knowledge
- Familiarity with internet usage and logical thinking skills

**Course Objectives:**

- To introduce the concept of Internet, Networks and its working principles.
- To know scripting languages and understand various applications related to Information Technology.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Apply website creation principles using various authoring tools effectively. |
| **CO2** | Utilize PHP scripting for dynamic web content and functionality. |
| **CO3** | Implement network configurations and protocols for efficient communication. |
| **CO4** | Apply mobile communication concepts to cellular network technologies. |
| **CO5** | Develop interactive and multimedia applications for different information systems. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Web and Internet Fundamentals:** Creating a Website – Working Principle of a Website, Browser Fundamentals – Types of Servers: Application Server, Web Server, Database Server, Introduction to Authoring Tools | 6 Hrs |
| 2. | **Client-Side and Server-Side Scripting Essentials:** Need for Scripting Languages – Types of Scripting Languages, Client-Side Scripting Overview, Server-Side Scripting with PHP – Working Principle of PHP, PHP Variables – Constants – Operators – Flow Control and Looping | 6 Hrs |
| 3. | **Advanced PHP and Web Interaction:** PHP Arrays – Strings – Functions – File Handling, PHP and HTML Integration – PHP and MySQL, Cookies – Simple PHP Scripts | 6 Hrs |
| 4. | **Networking Fundamentals:** Fundamental Computer Network Concepts, Types of Computer Networks – Network Layers – TCP/IP Model, Ethernet – Wi-Fi – Wireless Local Area Network, Network Components – Routing and Switching | 6 Hrs |

| | | |
|---|---|---|
| 5. | **Mobile Communication Technology**<br>Cell Phone Working Fundamentals – Frequencies and Channels, Digital Cell Phone Components – Generations of Cellular Networks, Mobile Network Technologies and Architecture, Voice Calls and SMS Fundamentals | 6 Hrs |
| 6. | **Application Development Essentials:**<br>Creation of Simple Interactive Applications, Simple Database Applications – Multimedia Applications, Design and Development of Information Systems, Personal Information System – Information Retrieval System – Social Networking Applications | 6 Hrs |
| | **TOTAL** | 36 Hrs |

**Text Books:**

1. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5" Third Edition, O'REILLY, 2014.
2. James F. Kurose, ―Computer Networking: A Top-Down Approach‖, Sixth Edition, Pearson, 2012.

**Reference Books:**

1. Gottapu Sasibhushana Rao, "Mobile Cellular Communication", Pearson, 2012.
2. R. Kelly Rainer , Casey G. Cegielski , Brad Prince, Introduction to Information Systems, FifthEdition, Wiley Publication, 2014.

**E-Resources:**

- NPTEL Online Courses: https://nptel.ac.in
- GeeksforGeeks: https://www.geeksforgeeks.org
- W3Schools : https://www.w3schools.com

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML) | | | | | | | | **Semester:** III | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Management Information System | | | | | | | | **Code: AMMC301** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **IE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 01 | - | 01 | 02 | - | - | 25 | - | - | 25 |

**Prerequisites:**

Basic knowledge of computer fundamentals and business operations.

**Course Objectives:**

- To Introduce the role of MIS in decision-making and business strategy.
- To Explore various types of information systems and their applications in organizations.
- To Explain the relationship between IT, organizations, and business processes.
- To Discuss contemporary technologies like ERP, BI, CRM, cloud computing, and analytics.
- To Analyze security, privacy, and ethical aspects in information systems.
- To Develop analytical and implementation skills using practical MIS tools.

**Course Outcomes:**

| | |
|---|---|
| **CO1** | Understand the foundational concepts and role of MIS in organizations. |
| **CO2** | Explain the types of information systems and their application in business functions. |
| **CO3** | Analyze the role of IT infrastructure in MIS including databases, networks, and cloud computing. |
| **CO4** | Evaluate enterprise applications like ERP, CRM, and SCM and their impact on business processes. |
| **CO5** | Understand decision-making processes, business intelligence, and data analytics in MIS. |
| **CO6** | Examine current trends, security, privacy, ethical issues, and implement MIS-based lab exercises. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to MIS**<br>Definition, Components, and Characteristics of MIS, Role of MIS in organizations and managerial decision-making, Strategic use of information systems for competitive advantage, Information system components: Hardware, Software, People, Process, Data. Types of decisions and levels of management<br>Case study: MIS implementation in enterprises | 6 Hrs |
| 2. | **Types of Information Systems**<br>Transaction Processing Systems (TPS), Management Information Systems (MIS), Decision Support Systems (DSS), Executive Support Systems (ESS), Enterprise Resource Planning (ERP) systems, Office Automation Systems (OAS), Functional | 6 Hrs |

| | | |
|---|---|---|
| | Business Systems: Marketing, HRM, Finance, Manufacturing, E-Business and E-Commerce systems | |
| 3. | **IT Infrastructure & Database Management**<br>Basics of IT infrastructure and computing platforms, Cloud computing and virtualization, Introduction to Database Management Systems (DBMS), Data warehousing and data marts. Mobile computing and IoT in MIS | 4 Hrs |
| 4. | **Enterprise Applications**<br>Enterprise Resource Planning (ERP): Features, Modules, Benefits. Customer Relationship Management (CRM): Architecture and Applications. Supply Chain Management (SCM): Functions and Impact. Integration of ERP, CRM, and SCM. Business Process Reengineering (BPR) and automation, Challenges in enterprise application implementation | 4 Hrs |
| 5. | **Decision Support & Business Intelligence**<br>Decision-making process and role of MIS in decision support, Business Intelligence (BI): Tools and Technologies, Data Mining and Big Data Analytics, Dashboards and Visualization tools, Introduction to Artificial Intelligence in MIS, Case Study: BI in retail, banking, or healthcare | 4 Hrs |
| 6. | **Security, Ethics, and Emerging Trends**<br>IT Security: Threats, vulnerabilities, and risk management. Security tools: Firewalls, IDS, encryption, authentication, Ethical and social issues in Information Systems. Legal aspects: Data protection laws (e.g., GDPR), Green computing, sustainability, and IT governance. Recent trends: Cloud MIS, MIS-as-a-Service, Blockchain in MIS | 4 Hrs |
| | **TOTAL** | 28 Hrs |

**Text Books:**

1. Kenneth C. Laudon, Jane P. Laudon, Management Information Systems: Managing the Digital Firm, Pearson, ISBN: 9781292342482
2. W.S. Jawadekar, Management Information System, Tata McGraw Hill, ISBN: 9780070263710

**Reference Books:**

1. James A. O'Brien, George Marakas, Management Information Systems, McGraw-Hill, ISBN: 9780073376813
2. Raymond McLeod, George Schell, Management Information Systems, Pearson, ISBN: 9780131408821
3. Ralph Stair, George Reynolds, Principles of Information Systems, Cengage Learning, ISBN: 9781337101910

**E-Resources:**

1. https://nptel.ac.in/courses/110105148
2. https://nptel.ac.in/courses/110105083

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | **Semester:** III |
|---|---|
| **Course:** Management Information System Lab | **Code: AMMC301** |

**Guidelines for Instructor's Manual**

The instructor's manual should briefly explain the course relevance to decision-making in AI & ML contexts. It must include course objectives, CO mapping with assignments, session-wise outcomes, tools/platforms to be used (e.g., MS Excel, Google Sheets, Power BI), and evaluation rubrics. Provide assignment examples with screenshots or reports. Focus on real-time problem-solving, automation potential, and business analytics integration.

**Guidelines for Student's Lab Journal**

The student lab journal should include:

- Cover page with student/institution details
- Index page
- Each experiment should include: Aim, Objectives, Theory, Tools Used, Procedure/Steps, Screenshots of execution (Excel, dashboards, scripts), Output, and Conclusion.
- Signature columns for each practical
- A final reflection or summary page

**Guidelines for Lab /TW Assessment**

- Assessment is **continuous** and based on timely submission, active lab participation, quality of reports/dashboards/scripts, and understanding during viva.
- Each practical should be evaluated out of 10 marks (5 for execution, 3 for explanation, 2 for documentation).
- Final Term Work (TW) should be calculated out of 25 marks.

**Guidelines for Practical Examination**

- **Structure:** The practical exam will consist of a problem-solving task related to the MIS concepts taught in the course.
- **Execution:** Students must complete the task in a given time frame, ensuring the use of relevant MIS tools and concepts.
- **Viva:** Post-execution, students will be questioned on the approach, tools used, and results obtained.
- **Evaluation:** Marks will be awarded based on the problem-solving ability, execution accuracy, and conceptual understanding during the viva.

**Guidelines for Laboratory Conduction**

- **Pre-Lab Briefing:** A short theoretical session should precede the practical to discuss key concepts and explain the practical goals.
- **Hands-On Work:** Encourage students to work on the tasks independently, while offering assistance when needed.

- **Post-Lab Discussion:** After completion, engage students in discussions about their approach, mistakes, and improvements.
- **Online Tools:** Consider using virtual labs, management simulation tools, or MIS software for practical sessions where applicable

**List of Assignments:- (Perform Any 10 Practical's)**

**Assignment 1:**
Introduction to Management Information Systems (MIS) and its applications. Install and configure an MIS tool (e.g., ERP, CRM) and demonstrate basic functionality.

**Assignment 2:**
Data Collection and Data Entry Techniques: Use an online database or software to enter, update, and manage information.

**Assignment 3:**
Design and implement a relational database using SQL for a small business. Create tables, relations, and queries for various MIS reports.

**Assignment 4:**
Create and execute a simple report generation using an MIS tool (e.g., sales report, inventory report). Learn how to interpret the generated reports.

**Assignment 5:**
Use an MIS tool to perform Data Analytics. Import data (e.g., sales or inventory data), apply basic analytics, and generate insights.

**Assignment 6:**
Develop a Dashboard using an MIS system to visualize key metrics such as sales, inventory levels, or customer satisfaction.

**Assignment 7:**
Use an MIS tool to simulate decision-making in business. Create scenarios based on financial reports to make decisions such as pricing, staffing, or resource allocation.

**Assignment 8:**
Implement a simple Customer Relationship Management (CRM) system using an MIS tool to track customer data, interactions, and sales performance.

**Assignment 9:**
Business Process Modeling: Use MIS software to model business processes and workflows in a business scenario. (e.g., supply chain, order processing)

**Assignment 10:**
Integrate external data into an MIS system. For example, import data from Excel or another external database and analyze it within the MIS tool.

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

**Assignment 11:**
Implement and demonstrate Inventory Management in an MIS system. Track items, quantities, suppliers, and costs, and generate reports for stock levels.

**Assignment 12:**
Case Study: Design an MIS solution for a small business or organization. Discuss system requirements, database design, tools to be used, and workflow management.

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** III | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Programming Lab-I | | | | | | | | **Code: AMVS303** | |
| **Teaching Scheme (Hrs/week)** | | | | **Evaluation Scheme (Marks)** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| - | 04 | - | 02 | - | - | 25 | - | 25 | 50 |

**Prerequisites:**

Basic understanding of programming logic, flowcharts, and pseudocode.
Fundamental knowledge of data types, variables, and operators

**Course Objectives:**

1. To introduce the concepts of object-oriented programming using C++.
2. To understand the syntax and structure of C++ programs.
3. To develop problem-solving skills by applying C++ programming constructs.
4. To learn the implementation of classes, objects, constructors, and various OOP features.
5. To explore advanced C++ concepts like operator overloading, inheritance, and exception handling.
6. To gain hands-on experience in memory management and data structure implementation using C++

**Course Outcomes:**

| | |
|---|---|
| **CO1** | Write, compile, and execute basic C++ programs for simple problem-solving |
| **CO2** | Implement object-oriented programming principles like classes, objects, and inheritance in C++ |
| **CO3** | Develop programs that perform complex operations such as operator overloading and memory Management. |
| **CO4** | Apply dynamic memory allocation and exception handling techniques in C++ programs |
| **CO5** | Solve real-world problems by implementing matrix operations and other complex data Structures using C++. |
| **CO6** | Demonstrate the use of pointers, constructors, and destructors in C++ applications |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Basics of C++ Programming:**<br>Introduction to C++ syntax, Input/Output operations.<br>Control structures: if-else, switch-case, loops (for, while, do-while).<br>Arrays and string handling in C++ | 9 |
| 2. | **Basic Problem-Solving with C++:**<br>Number reversal and factorial calculation, Prime number generation<br>Finding largest and smallest elements in a list. | 9 |
| 3. | **Dynamic Memory Allocation and Sorting:**<br>Arrays and dynamic memory management using new and delete.<br>Sorting algorithms and their implementation in C++. | 9 |

| | | |
|---|---|---|
| 4. | **Object-Oriented Programming:**<br>Class declaration and member functions.<br>Constructors: default, parameterized, and copy constructors. | 9 |
| | Implementing a class STUDENT with data members and member functions. | |
| 5. | **Advanced Concepts in C++:**<br>Operator overloading (unary and binary).<br>Function overloading and friend functions.<br>Pointers and dynamic object access. | 10 |
| 6. | **Inheritance, Exception Handling, and Matrix ADT:**<br>Inheritance: Single, Multiple, Multilevel, Hierarchical.<br>Matrix operations using the Matrix ADT class.<br>Exception handling and constructors/destructors. | 10 |
| | **TOTAL** | **56** |

**List of Experiments: (**Any Nine experiments from list 1 to 11 are mandatory and an experiment 12 is mandatory)

| | |
|---|---|
| 1 | **Introduction to C++ Programming**<br>Write basic C++ programs demonstrating:<br>● Input/Output operations<br>● Class and object definitions<br>● Control statements (if-else, switch-case)<br>● Looping (for, while, do-while)<br>● Array manipulation<br>● String handling |
| 2 | Write a C++ program to calculate the factorial of a given number |
| 3 | Write a C++ program to generate all prime numbers between 1 and n, where n is provided by the user. |
| 4 | **Sorting and Dynamic Memory Allocation: (Any one)**<br>a) Write a C++ program to sort a list of numbers in ascending order.<br>b) Write a C++ program to illustrate dynamic memory allocation using the new and delete keywords |
| 5 | **Class Definitions and Constructors:**<br>a) Write a C++ program illustrating class declaration, definition, and member access.<br>b) Write a C++ program to demonstrate the use of default, parameterized, and copy constructors. |
| 6 | **Implementing a Class STUDENT**<br>a) Write a C++ program to implement a class **COURSE** with the following members:<br>**Data Members:**<br>● **CourseName**: Name of the course<br>● **CourseCode**: Unique code for the course<br>● **Credits**: Credits assigned to the course<br>● **StudentList[]**: Array to store names of enrolled students<br>● **MaxStudents**: Maximum number of students that can enroll in the course |

| | |
|---|---|
| 7 | **Operator and Function Overloading:**<br>a) Write a C++ program demonstrating:<br>● Operator overloading for unary and binary operators.<br>● Function overloading for multiple function definitions.<br>b) Write a C++ program to demonstrate friend functions and friend classes. |
| 8 | **Pointers and Object Access**<br>a) Write a C++ program to dynamically allocate memory for an object of the class **EMPLOYEE** and access its members using pointers.<br>b) Write a C++ program to generate a Fibonacci series using a constructor to initialize data Members. |
| 9 | **Matrix ADT Implementation**<br>Write a C++ program to implement a matrix Abstract Data Type (ADT) using a class. The operations supported by this ADT are:<br>● Reading a matrix<br>● Addition of matrices<br>● Subtraction of matrices<br>● Printing a matrix. |
| 10 | **Inheritance in C++:**<br>Write C++ programs to demonstrate the following forms of inheritance:<br>(a) Single Inheritance<br>(b) Multiple Inheritance<br>(c) Multilevel Inheritance<br>(d) Hierarchical Inheritance |
| 11 | **Constructors and Base Class Pointer**<br>a) Write a C++ program to illustrate the order of execution of constructors and destructors when a new class is derived from more than one base class.<br>b) Write a C++ program to invoke derived class members through a base class pointer. |
| 12 | **Exception Handling in C++:**<br>a) Write a C++ program containing a possible exception. Use a try block to throw the Exception and a catch block to handle it.<br>b) Write a C++ program to demonstrate catching all exceptions using generic exception Handlers. |

**Reference Books:**

1. "Programming: Principles and Practice Using **C++**" by Bjarne Stroustrup
2. "Effective C++: 55 Specific Ways to Improve Your Programs and Designs" by Scott Meyers
3. "Accelerated C++: Practical Programming by Example" by Andrew Koenig and Barbara E. Moo

**E-Resources:**

1. C++ Programming Tutorial (NPTEL)

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** III | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Mini Project | | | | | | | | **Code: AMCE301** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| - | 02 | - | 01 | - | - | 25 | - | - | 25 |

**Preamble:**

Project Based Learning (PBL) is a dynamic teaching methodology that fosters critical thinking, collaboration, and personalized learning experiences. In this approach, students work in groups to explore topics of personal interest through meaningful inquiry. Projects are rooted in real-world problems, aligned with the curriculum, and often span multiple disciplines. Students take ownership of their learning by determining how to approach the problem and selecting appropriate methods and activities. They gather information from diverse sources, analyze and synthesize data, and derive meaningful insights. The real-life relevance of PBL enhances student motivation and lends authenticity to their work, while also building essential life skills like teamwork and self-reflection. Each group is supported by a faculty member, known as a mentor, who provides guidance throughout the process. Technology plays a vital role in facilitating collaboration and supporting each phase of the project. At the conclusion of the project, students present their outcomes, demonstrating both their knowledge and communication skills. They also engage in self-assessment to reflect on their progress and learning. The mentor's role is to support and advise, rather than to control or manage, encouraging student autonomy and deeper engagement.

**Companion Course:**

Online courses relevant to the project, along with expert lecture on Intellectual property rights, patents and software engineering.

**Course Objectives :**

- To understand the key processes and methodologies involved in Project-Based Learning.
- To enhance students' critical thinking abilities and their skills in solving real-world engineering problems.
- To illustrate the roles and responsibilities of IT engineers in addressing engineering challenges within social, environmental, and economic contexts.
- To prepare students with the knowledge and practical skills needed to develop effective solutions for problems typically presented in hackathons.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Develop innovative solutions for real-life problems by applying analytical thinking and collaborative problem-solving. |
| **CO2** | Demonstrate the ability to learn through hands-on experiences, fostering habits of lifelong learning. |

| | |
|---|---|
| **CO3** | Address and resolve technical challenges in real-world scenarios through effective teamwork. |
| **CO4** | Participate actively in interdisciplinary collaborations to enhance knowledge integration and practical understanding. |

**COURSE CONTENTS**

**Group Structure**

Students will work collaboratively in **mentor-guided teams** to plan, manage, and complete projects or activities that address defined real-world problems.

1. Each team should consist of **3 to 6 students**, working cohesively toward a common goal.
2. A **dedicated mentor** will be assigned to each group to support their learning and development throughout the PBL journey.

**Project/Problem Selection**

1. Projects may originate from **any domain,** though topics with an **IT or technical focus are encouraged.**
2. Projects initiated in the **first year of engineering** may be extended if they show potential and relevance.
3. Preference should be given to projects that involve **conceptual modeling** and the **application of software tools.**
4. Multiple solution approaches — such as **theoretical analysis, practical implementation, working models, simulations,** or software-based solutions — are encouraged.
5. Projects that require a **multidisciplinary perspective** are highly desirable.
6. Problems should demand **in-depth exploration** of practical, scientific, or technical areas.
7. The project methodology should incorporate **hands-on activities, organizational and field visits, interaction with research institutes**, and **expert consultations** to expose students to the latest technologies and practices.

**Assessment**

The department shall ensure rigorous **monitoring and assessment** of both student progress and the impact of their proposed solutions.

- **Weekly reviews** will be conducted to track the project's progress.
- Assessment will include both **individual and group performance**, facilitated by the assigned mentor.
- Students are expected to maintain a culture of **authentic collaboration, self-motivation, and peer learning.**
- Departments must provide **guidance, orientation, and access to resources** to support the project process.
- Students will showcase their learning through **public products, reports, or presentations.**

***Components of Assessment:***

1. **Individual Evaluation** – Assessment of personal involvement, contribution, and understanding.
2. **Group Evaluation** – Review of team dynamics, role distribution, communication, and cohesion.

3. **Presentation and Documentation** – Evaluation of reporting quality, organization, and communication of outcomes.

## Evaluation and Continuous Assessment

Students and mentors are required to maintain a **PBL workbook** that documents progress, deliverables, and reflections.

- The workbook will serve as a record of **accountability, punctuality, technical writing skills,** and project workflow.
- A **Continuous Assessment Sheet (CAS)** will be maintained by mentors and the department.

***Suggested Assessment Criteria and Weightage:***

1. **Idea Inception** – 5%
2. **Solution Quality / Problem Solving / Final Output** – 40%
3. **Documentation (requirements, design, tools, report, etc.)** – 25%
4. **Patentability Potential** – 10%
5. **Demonstration (UI, usability, presentation)** – 10%
6. **Participation in Competitions / Publications** – 5%
7. **Consideration of Environmental, Ethical, Legal, or Social Aspects** – 5%

**Rubrics** for evaluation should be designed based on these criteria and shared with students for transparency and guidance.

## Faculty / Mentor is expected to perform following activities

Faculty mentors are expected to actively support student learning by:

- Revising PBL concepts and assessing initial student skills
- Forming diverse and balanced teams
- Sharing knowledge on IPR (patents, copyrights, publications)
- Discussing case studies to provide practical insight
- Designing and explaining the rubrics for evaluation
- Conducting weekly assessments of deliverables (presentations, reports, concept maps, logbooks)
- Providing continuous guidance (scaffolding)
- Engaging in both formative and summative assessment practices

## Reference Books:

1. Project-Based Learning, Edutopia, March 14,2016.
2. What is PBL? Buck Institute for Education.
3. www.schoology.com
4. www.wikipedia.org
5. www.howstuffworks.com

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | **Semester:** III | | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Internship-II | | | | | | | **Code: AMIN302** | | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 4 Week | | | 02 | | - | 25 | - | - | 25 |
| | | | | | | | | | |
| Kindly refer the policy of internship on College Website | | | | | | | | | |

# Semester-IV

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | **Semester:** IV | | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Database Management System | | | | | | | **Code: AMPC405** | | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 03 | 02 | - | 04 | 40 | 60 | - | - | 50 | 150 |

**Prerequisites:**

- Basic knowledge of data structures and programming
- Understanding of file systems and operating system fundamentals

**Course Objectives:**

- To introduce the fundamental concepts of database systems
- To design relational databases using E-R models and normalization techniques
- To develop skills in SQL for managing and querying relational databases
- To understand transaction processing, concurrency control, and recovery
- To explore indexing, query optimization, and database architectures
- To gain insight into emerging trends like NoSQL and distributed databases

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Understand basic database concepts and architecture |
| **CO2** | Design relational databases using E-R models and normalization |
| **CO3** | Write SQL queries to manage, retrieve, and manipulate data |
| **CO4** | Apply transaction management, concurrency control, and recovery techniques |
| **CO5** | Analyze indexing strategies and query optimization methods |
| **CO6** | Explore advanced topics such as NoSQL, distributed, and cloud-based databases |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Database Systems**<br>Purpose of databases, file system vs DBMS, advantages of DBMS, database users and administrators, DBMS architecture, data models (hierarchical, network, relational), schema and instance, data independence, DBMS languages (DDL, DML). | 6 Hrs |
| 2. | **Entity-Relationship (ER) Modeling and Relational Model**<br>ER model: entity types, attributes, relationships, generalization, specialization, aggregation. Converting ER to relational schema. Relational model: structure, keys, integrity constraints, relational algebra operations. | 6 Hrs |

| | | |
|---|---|---|
| 3. | **Structured Query Language (SQL)**<br>Basic SQL commands (SELECT, INSERT, DELETE, UPDATE), constraints, aggregate functions, group by, order by, joins (inner, outer, self), subqueries, views, indexes, and stored processes. | 6 Hrs |
| 4. | **Normalization and Functional Dependencies**<br>Functional dependencies, decomposition, lossless-join and dependency preservation, normal forms (1NF, 2NF, 3NF, BCNF), multivalued dependencies, join dependencies, denormalization. | 6 Hrs |
| 5. | **Transaction Management and Concurrency Control**<br>Transaction properties (ACID), transaction states, serializability, conflict and view serializability, concurrency control protocols: locking, timestamp ordering, deadlock handling, recovery concepts and techniques. | 6 Hrs |
| 6. | **Query Optimization and Advanced Topics**<br>Query processing overview, evaluation strategies, indexing (B+ trees, hash-based), and introduction to NoSQL, distributed databases, cloud-based databases, and emerging trends in database systems. | 6 Hrs |
| | **TOTAL** | 36 Hrs |

**Text Books:**

- Korth, Silberschatz & Sudarshan, *Database System Concepts*, McGraw Hill Education
- Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson Education

**Reference Books:**

1. C.J. Date, *An Introduction to Database Systems*, Addison-Wesley
2. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, McGraw Hill
3. Rob and Coronel, *Database Systems: Design, Implementation, and Management*, Cengage Learning

**E-Resources:**

NPTEL – Database Management Systems: https://nptel.ac.in/courses/106/105/106105175/

1. GeeksforGeeks – DBMS: https://www.geeksforgeeks.org/dbms/
2. Oracle SQL Tutorial: https://docs.oracle.com/en/database/

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | **Semester:** IV |
|---|---|
| **Course:** Programming Lab 2 | **Code: AMVS404** |

**Guidelines for Instructor's Manual**

The instructor's manual should comprehensively describe the lab's objectives, university guidelines, mapping of each experiment to the course outcomes, and a week-wise conduction plan. It should include conceptual overviews of each topic, sample SQL commands, expected results, and evaluation rubrics to ensure uniformity across batches. The manual should help faculty members guide students in understanding the practical implementation of theoretical DBMS concepts through hands-on lab assignments and case studies using open-source tools such as MySQL.

**Guidelines for Student's Lab Journal**

The student's lab journal is a formal academic document and should reflect systematic learning and execution. Each practical entry must include the aim of the experiment, problem definition, step-by-step approach, SQL queries or PL/SQL code, and screenshots or typed outputs. It should conclude with brief observations and learnings. The journal should begin with a cover page, certificate of completion, and table of contents. It must be maintained regularly and signed weekly by the concerned faculty, reflecting the student's progressive understanding of the subject.

**Guidelines for Lab /TW Assessment**

Term work evaluation should be continuous, based on regular attendance, timely completion of practical's, correctness of SQL syntax, query output, and quality of documentation in the journal. Additional weightage may be given to innovative approaches or enhanced versions of the given assignments. Every experiment should be mapped with specific course outcomes to ensure skill development. The assessment should also account for the student's performance in the viva conducted during lab sessions.

**Guidelines for Practical Examination**

The practical examination should consist of one SQL or PL/SQL problem covering key concepts from the syllabus such as joins, queries, normalization, triggers, or stored procedures. Students should implement the assigned task and explain their approach during the viva. Evaluation should be based on logic, correctness, adherence to normalization principles, schema design, and clarity of understanding during the oral exam. Students may also be asked to draw an ER diagram or explain transaction control concepts.

**Guidelines for Laboratory Conduction**

Each lab session should begin with a brief explanation of the theory behind the day's assignment, followed by demonstration and individual coding. Faculty should monitor student progress actively, promote best practices in SQL programming, and encourage the use of schema planning and ER diagrams before implementation. Student queries must be addressed interactively, and consistent support should be offered to ensure conceptual clarity and application readiness.

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **List of Assignments:- (Perform Any 10 Practical's)** |
|---|
| 1. **Basic SQL Commands — CO1** |
| Execute SQL commands to create, insert, update, delete, and retrieve data from a single table. Apply integrity constraints (primary key, not null, unique, default, check). |
| 2. **SQL with Multiple Tables and Joins — CO2** |
| Perform queries involving two or more tables using INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. Retrieve complex data using aliasing and filtering. |
| 3. **Aggregate Functions and Grouping — CO2** |
| Write SQL queries using aggregate functions (SUM, AVG, COUNT, MIN, MAX) along with GROUP BY, HAVING, and ORDER BY. |
| 4. **Nested Queries and Subqueries — CO2** |
| Use nested subqueries in SELECT, FROM, and WHERE clauses. Demonstrate use of correlated subqueries and EXISTS operator. |
| 5. **Views and Indexing — CO3** |
| Create and use views to simplify complex queries. Create indexes on selected columns to improve performance. Analyze execution plans if supported by the DBMS. |
| 6. **ER Modeling and Table Creation — CO2** |
| Design an ER model for a real-world scenario (e.g., Library or Hospital). Convert ER into relational schema and implement the schema using SQL CREATE TABLE. |
| 7. **Normalization — CO3** |
| Given a data set or set of attributes, normalize it into 1NF, 2NF, 3NF, and BCNF. Identify and explain functional dependencies and keys. |
| 8. **Stored Procedures and Functions — CO4** |
| Create stored procedures and functions using PL/SQL or equivalent in MySQL/PostgreSQL. Demonstrate passing parameters and returning values. |
| 9. **Triggers and Cursors — CO4** |
| Create BEFORE and AFTER triggers for insert/update/delete operations. Write cursors to iterate through query results and perform row-wise operations. |
| 10. **Transaction Control and Concurrency — CO5** |
| Simulate multiple users performing transactions. Demonstrate commit, rollback, and savepoint. Handle concurrency using isolation levels and locking. |

| |
|---|
| 11. **Case Study: Real-Time Database Application — CO6** |
| Choose a real-world system (e.g., Student Portal, Inventory, Hospital). Model ER diagram, create schema, write at least 5–6 queries, and explain relationships and normalization. |
| 12. **Mini Project: Database-Driven Application — CO6** |
| Design and implement a complete application (e.g., Online Bookstore, Hotel Booking System). Include ER diagram, schema creation, query examples, and optional frontend using any GUI or web interface. Submit code, documentation, and screenshots. |
| **Text Books:** |
| • Korth, Silberschatz & Sudarshan, *Database System Concepts*, McGraw Hill Education<br>• Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson Education |
| **Reference Books:** |
| 1. C.J. Date, *An Introduction to Database Systems*, Addison-Wesley<br>2. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, McGraw Hill<br>3. Rob and Coronel, *Database Systems: Design, Implementation, and Management*, Cengage Learning |
| **E-Resources:** |
| 1. NPTEL – Database Management Systems: https://nptel.ac.in/courses/106/105/106105175/<br>2. GeeksforGeeks – DBMS: https://www.geeksforgeeks.org/dbms/<br>3. Oracle SQL Tutorial: https://docs.oracle.com/en/database/ |

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML) | | | | | | | | **Semester:** IV | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Data Science | | | | | | | | **Code: AMPC406** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **IE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 03 | 02 | - | 04 | 40 | 60 | - | - | 25 | 125 |

**Prerequisites:**

Python Programming, Statistics, Linear Algebra

**Course Objectives:**

- To introduce the core concepts, methodologies, and tools of data science.
- To equip students with the skills to analyze, visualize, and interpret complex data.
- To apply statistical and machine learning models to solve real-world problems.
- To develop proficiency in tools like Python, pandas, NumPy, scikit-learn, and Jupyter.
- To build, evaluate, and deploy data-driven models and communicate insights effectively.

**Course Outcomes:**

| | |
|---|---|
| **CO1** | Understand the fundamentals and lifecycle of data science with data types and sources. |
| **CO2** | Perform data cleaning, preprocessing, and exploratory data analysis using Python. |
| **CO3** | Apply statistical analysis and hypothesis testing for data inference. |
| **CO4** | Build and evaluate machine learning models for supervised and unsupervised learning. |
| **CO5** | Utilize advanced techniques such as ensemble learning, feature engineering, NLP, and time series. |
| **CO6** | Work on end-to-end data science projects using real-world datasets and deploy solutions. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Data Science**<br>What is Data Science? Roles of a data scientist, Data Science process/lifecycle, Data types: structured, semi-structured, unstructured. Data sources: APIs, Web scraping, databases, files. Introduction to Python for Data Science: Jupyter, NumPy, pandas, Working with data frames, basic operations | 6 Hrs |
| 2. | **Data Wrangling, Cleaning, and Visualization**<br>Handling missing data, duplicates, and outliers, Data normalization and transformation, Feature scaling, encoding categorical variables, Data visualization tools: matplotlib, seaborn, plotly, EDA techniques: histograms, boxplots, heatmaps, pairplots, Data imbalance handling (SMOTE, undersampling) | 6 Hrs |
| 3. | **Statistical Analysis and Hypothesis Testing** | 6 Hrs |

| | | |
|---|---|---|
| | Descriptive statistics: measures of central tendency & dispersion, Probability distributions: Normal, Binomial, Poisson, Inferential statistics: Sampling, confidence intervals, Hypothesis testing: t-test, ANOVA, chi-square, A/B Testing & its applications in real-world scenarios | |
| 4. | **Machine Learning for Data Science**<br>Machine learning workflow, Supervised learning: Linear regression, logistic regression, decision trees, random forest. Model evaluation: accuracy, precision, recall, F1-score, confusion matrix, ROC-AUC. Unsupervised learning: k-means, hierarchical clustering, PCA, Model selection and cross-validation | 6 Hrs |
| 5. | **Real-time Data Processing & MLOps**<br>Real-time Data and Stream Processing, Introduction to streaming data, Data Pipelines and Workflow Automation, Batch vs stream processing pipelines, Data pipeline design principles. Introduction to MLOps, MLOps lifecycle: model versioning, CI/CD for ML, MLFlow: tracking experiments, model registry. Data versioning with DVC, Model Deployment, Serving models via REST APIs using FastAPI or Flask, Model containerization with Docker | 6 Hrs |
| 6. | **Advanced Data Science Topics & Responsible AI**<br>Feature Engineering, Feature transformation, binning, interactions. Feature selection: filter, wrapper, and embedded methods. Time Series Analysis, Components: trend, seasonality, noise. Forecasting models: ARIMA, Holt-Winters, Facebook Prophet, Model Explainability and Interpretability, SHAP, LIME, Partial Dependence Plots (PDPs). Ethics in AI and Data Science, Bias in datasets and models, Fairness, transparency, and accountability, Introduction to responsible AI frameworks (like IBM AI Fairness 360, Google PAIR) | 6 Hrs |
| | **TOTAL** | 36 Hrs |

**List of Experiments:**

1) Exploring DataFrames with Pandas
Load and manipulate datasets using pandas, including filtering, sorting, indexing, and grouping.
2) Reading and Writing Files (CSV, JSON, Excel)
Perform operations on various data formats and convert between them.
3) Handling Missing Data and Outliers
Apply imputation techniques, detect outliers using IQR/Z-score, and visualize distributions.
4) Feature Encoding and Scaling
Apply label encoding, one-hot encoding, and scale features using normalization and standardization.
5) Exploratory Data Analysis using Seaborn & Plotly
Generate univariate, bivariate, and multivariate visualizations to derive insights.
6) Descriptive Statistics and Probability Distributions
Compute mean, median, mode, standard deviation; plot normal and Poisson distributions.
7) Hypothesis Testing (t-test, ANOVA)
Perform tests using scipy.stats and interpret results for real-world problems.
8) Linear Regression and Polynomial Regression
Predict continuous values and evaluate models using $R^2$, MSE.
9) Classification with Logistic Regression & Decision Trees
Train, test, and evaluate models using confusion matrix, precision, recall, and ROC curves.

10) Clustering using K-Means and Hierarchical Clustering
Segment customers/data points and visualize clusters.
11) Dimensionality Reduction using PCA
Apply PCA and visualize high-dimensional data in 2D.
12) Building a Real-time Pipeline using Kafka or Spark Streaming
Simulate real-time data flow from a producer to a consumer with basic transformations.
13) Tracking ML Experiments using MLflow
Log parameters, metrics, and save model versions.
14) Model Deployment using FastAPI or Flask
Build a REST API for serving a trained ML model.
15) Containerizing ML Model with Docker
Package and run an ML model inside a Docker container.
16) Creating Dashboards using Streamlit or Power BI
Build an interactive dashboard for a dataset with filters and visual summaries.
17) Using Google Colab and BigQuery for Scalable Analysis
Query public datasets using SQL and analyze results in Python.
18) Mini Capstone Project
End-to-end project: data collection, cleaning, modeling, deployment, and reporting.

**Text Books:**

3. Aurélien Géron, Hands-On ML with Scikit-Learn, Keras & TensorFlow, O'Reilly, ISBN 9781492032649
4. Wes McKinney, Python for Data Analysis, O'Reilly, ISBN 9781491957660

**Reference Books:**

4. Cathy O'Neil, Rachel Schutt, Doing Data Science, O'Reilly, ISBN 9781449358655
5. Joel Grus, Data Science from Scratch, O'Reilly, ISBN 9781492041139
6. Jake VanderPlas, Python Data Science Handbook, O'Reilly, ISBN 9781491912058

**E-Resources:**

3. https://nptel.ac.in/courses/106106179
4. https://nptel.ac.in/courses/106106226

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | **Semester:** IV | | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Software Engineering | | | | | | | **Code: AMPC407** | | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 03 | - | - | 03 | 40 | 60 | - | - | - | 100 |

**Prerequisites:**

- Basic knowledge of the Internet and the World Wide Web
- Familiarity with business or commerce principles
- Interest in digital platforms and technologies

**Course Objectives:**

- Understand the role of software engineering in system development.
- Learn different software development life cycle models.
- Apply design methodologies, testing techniques, and quality assurance.
- Use project management tools and principles.
- Explore advanced practices like Agile, DevOps, and software security.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Define and explain the foundational principles of software engineering. |
| **CO2** | Identify and apply appropriate software process models. |
| **CO3** | Analyze and document software requirements using industry-standard techniques. |
| **CO4** | Design software using modeling tools and design patterns. |
| **CO5** | Apply software testing strategies and evaluate software quality. |
| **CO6** | Utilize modern engineering practices such as Agile, DevOps, and CI/CD pipelines. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **UNIT 1: Introduction to Software Engineering:**<br>Introduction to SE, Myths & Principle, Software Development Life Cycle (SDLC) Process Models: Waterfall, Incremental, Spiral, V-Model, Software Engineering Ethics (ACM/IEEE Code), Software vs. Program, Software Characteristics<br>Introduction to CASE Tools | 6 Hrs |
| 2. | **UNIT 2: Requirements Engineering & Feasibility Study**<br>Requirement Engineering Phases, Functional & Non-functional Requirements<br>**SRS:** Structure and IEEE Standard Format, Feasibility Analysis: Technical, Operational, Economic, Use Case Modeling and Requirement Tools | 6 Hrs |

| | | |
|---|---|---|
| 3. | **UNIT 3: Software Design & Modeling**<br>Principles of Design: Modularity, Abstraction, Coupling, Cohesion, Design Techniques: Structured and Object-Oriented, UML Diagrams: Use Case, Class, Sequence, Activity, Introduction to Design Patterns: Singleton, MVC | 6 Hrs |
| 4. | **UNIT 4: Software Testing and Maintenance**<br>Testing Lifecycle, Types of Testing (Unit, Integration, System), Testing Techniques: BVA, Equivalence Partitioning, Path Testing, White Box & Black Box Testing, Test Case Design & Test Plan Preparation, Software Maintenance, Configuration Management | 6 Hrs |
| 5. | **UNIT 5: Software Project Management**<br>Project Planning, Scheduling, Risk Management, Work Breakdown Structure (WBS), Gantt Charts, Cost Estimation Techniques (COCOMO, FP), Software Metrics: LOC, Cyclomatic Complexity, Tools: JIRA, Trello, GitHub Projects | 6 Hrs |
| 6. | **UNIT 6: Advanced Trends in Software Engineering**<br>Agile Software Development: Scrum, Kanban, DevOps Lifecycle, CI/CD Pipelines, Software Reengineering and Reverse Engineering, Software Security: OWASP Basics<br>**Case Study:** Modern Software Stack in Use (e.g., Netflix, Spotify) | 6 Hrs |
| | **TOTAL** | 36 Hrs |

**Text Books:**

1. Software Engineering: A Practitioner's Approach – Roger S. Pressman (McGraw Hill)
2. Software Engineering – Ian Sommerville (Pearson Education)

**Reference Books:**

- An Integrated Approach to Software Engineering – Pankaj Jalote (Narosa)
- Software Project Management – Bob Hughes, Mike Cotterell (McGraw Hill)
- Fundamentals of Software Engineering – Rajib Mall (PHI)

**Digital References:**

- NPTEL: Software Engineering - IIT KGP
- Coursera: Agile with Atlassian Jira (Atlassian)
- GeeksforGeeks: SE Portal
- GitHub Education: Student Developer Pack

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | | **Semester:** IV |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Knowledge Representation and Reasoning | | | | | | | | | **Code: AMMD402** |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 03 | - | - | 03 | 40 | 60 | - | - | - | 100 |

**Prerequisites:**

- fundamentals of Artificial Intelligence
- Basic Discrete Mathematics and Logic
- Understanding of algorithms and data structures

**Course Objectives:**

- To introduce the fundamental concepts of knowledge representation and inference.
- To explore logical formalisms used for modeling intelligent behavior.
- To understand rule-based systems, semantic networks, and ontologies.
- To develop reasoning techniques using propositional and predicate logic.
- To explore automated reasoning tools and their applications in AI.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Understand the role of knowledge representation in intelligent systems |
| **CO2** | Apply propositional and predicate logic for representing knowledge |
| **CO3** | Model real-world problems using semantic networks and ontologies |
| **CO4** | Design and implement rule-based expert systems |
| **CO5** | Analyze reasoning techniques such as forward and backward chaining |
| **CO6** | Utilize reasoning tools and inference engines in AI applications |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Knowledge Representation**: Meaning, need, and types of knowledge. Declarative vs. procedural knowledge. Overview of KR techniques: logic-based, semantic network, frames, rules, ontologies. | 6 Hrs |
| 2. | **Propositional Logic**: Syntax and semantics, truth tables, inference rules, logical equivalences, entailment, satisfiability. Applications of propositional logic in AI. | 6 Hrs |
| 3. | **First Order Predicate Logic (FOPL)**: Syntax, semantics, quantifiers, unification, resolution, knowledge base construction. Limitations of FOPL. | 6 Hrs |

| | | |
|---|---|---|
| 4. | **Rule-Based Systems and Expert Systems**: Production rules, forward chaining, backward chaining, conflict resolution strategies. Architecture of rule-based expert systems. | 6 Hrs |
| 5. | **Semantic Networks, Frames and Ontologies**: Conceptual graphs, inheritance, slot-filler structure, Description Logics, OWL basics, ontology engineering for AI systems. | 6 Hrs |
| 6. | **Non-monotonic Reasoning and Uncertainty**: Closed world assumption, default reasoning, belief revision. Introduction to probabilistic logic and fuzzy logic-based representation. | 6 Hrs |
| | **TOTAL** | 36 Hrs |

**Text Books:**

- Brachman, Ronald J., and Hector J. Levesque. *Knowledge Representation and Reasoning*, Elsevier, 2004.
- Elaine Rich and Kevin Knight, *Artificial Intelligence*, McGraw-Hill.
- Nils J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann.

**Reference Books:**

- Peter Jackson, *Introduction to Expert Systems*, Pearson Education
- John F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole
- Rajendra Akerkar, *Artificial Intelligence*, PHI Learning

**Digital References:**

- NPTEL Course: Artificial Intelligence: Knowledge Representation & Reasoning
- Stanford AI Materials: https://ai.stanford.edu
- YouTube Series: "AI – Knowledge Representation" by IIT Kharagpur
- MIT OCW: *Artificial Intelligence – KR Topics* [https://ocw.mit.edu]
- OWL & RDF Tutorials: https://www.w3.org/TR/owl-guide/

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML) | | | | | | | **Semester:** IV | | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Open Elective -II (Cyber Security & Laws) | | | | | | | **Code: ALOE402** | | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| **02** | **-** | **-** | **02** | **40** | **60** | **-** | **-** | **-** | **100** |

**Prerequisites:**

Foundational knowledge in computer science, networking.

**Course Objectives:**

- To introduce the cyber world and cyber law in general.
- To enhance the understanding of problems arising out of online transactions and provoke them to find solutions.
- To know the technologies that stand behind certain Cyber Crimes.
- To identify and analyze statutory, regulatory, constitutional, and organizational laws that affect the information technology professional.

**Course Outcomes:**

| | |
|---|---|
| **CO1** | Illustrate and discuss the basic concepts of cyber security, Devices. |
| **CO2** | Understand the aspects related to personal data privacy and security. |
| **CO3** | Understand the main components of a cyber security plan. |
| **CO4** | Understand about the type and nature of cyber-crimes. |
| **CO5** | Articulate the well-known cyber-attack incidents, explain the attack scenarios, and explain mitigation techniques. |
| **CO6** | Explain the difference between Systems Cyber Security, Network Cyber Security, and Cryptography, Crypto-Protocols, etc. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to Cyber security :**Defining Cyberspace and Overview of Computer and Web-technology, Architecture of cyberspace, Communication and web technology, Internet, World wide web, Internet infrastructure for data transfer and governance, Internet society, Regulation of cyberspace, Concept of cyber security. | 4 Hrs. |
| 2. | **Digital Devices Security, Tools and Technologies for Cyber Security :** End Point device and Mobile phone security, Password policy, Security patch management, Data backup, Downloading and management of third party software, Device security policy, Cyber Security best practices, | 4 Hrs. |

| | | |
|---|---|---|
| | Significance of host firewall, Wi-Fi security, Configuration of basic security policy and permissions. | |
| 3. | **Cyber security Management, Compliance and Governance:** Cyber security Plan- cyber security policy, cyber crises management plan., Business continuity, Risk assessment, Types of security controls and their goals, Cyber security audit and compliance, National cyber security policy and strategy. | 4 Hrs. |
| 4. | **Cyber crimes :** Targeting Computer systems and Mobiles- data diddling attacks, spyware, logic bombs, DoS, DDoS, APTs, virus, Trojans, ransomware, data breach Cybersquatting, Pharming, drug trafficking, human trafficking., Social Media Scams & Frauds- impersonation, identity theft Cyber Police stations, Crime reporting procedure, Case studies | 4 Hrs. |
| 5. | **Cybercrime**: Illustrations, Examples and Mini cases ,Real-life examples, Mini cases-Social Engineering Attacks, Illustration of financial frauds in cyber domain, Digital Signature- Related Crime scenarios, Digital forensics case Illustrations, Online scams**.** | 4 Hrs. |
| 6. | **Cyber law:** Cybercrime and legal landscape around the world, IT Act,2000 and its amendments. Limitations of IT Act, 2000. Cybercrime and punishments, Cyber Laws and Legal and ethical aspects related to new technologies- AI/ML, IOT, Block chain, Dark net and Social media, Cyber Laws of other countries. | 4 Hrs. |
| | **TOTAL** | 24 Hrs. |

**Text Books:**

1. Michael E. Whitman, Herbert J. Mattord, (2018). Principles of Information Security, 6th edition, Cenage Learning, N. Delhi.
2. Chris Reed & John Angel, Computer Law, OUP, New York, (2007).
3. Cyber Crime an Introduction by Prasad R.S.
4. Cyber Laws by Ed. Kumar Krishna.

**Reference Books:**

1. Angus M. Marshall, "Digital forensics: Digital evidence in criminal investigation", John – Wiley and Sons, 2008.
2. Sushma Arora, Raman Arora, Cyber Crimes & Laws, 4th Edition 2021, Publisher: Taxmann, ISBN-10: 9390712491
3. N S Nappinai, Technology Laws Decoded, 1st Edition, Publisher: Lexis Nexis, ISBN: 9789350359723

**E-Resources:-**

1. https://onlinecourses.nptel.ac.in/noc23_cs127/preview
2. https://onlinecourses.swayam2.ac.in/nou19_cs08/preview
3. https://onlinecourses.nptel.ac.in/noc25_cs116/preview

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** IV | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** E-Commerce | | | | | | | | **Code: AMMC402** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 01 | 01 | - | 02 | - | - | 25 | - | - | 25 |

**Prerequisites:**

- Basic knowledge of the Internet and the World Wide Web
- Familiarity with business or commerce principles
- Interest in digital platforms and technologies

**Course Objectives:**

- To provide an understanding of fundamental concepts and models in E-Commerce.
- To familiarize students with digital business strategies and payment systems.
- To develop awareness of legal, ethical, and security issues in E-Commerce.
- To give exposure to digital marketing and E-Commerce tools..

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Understand the foundations and evolution of E-Commerce and its business models. |
| **CO2** | Analyze various payment systems, technologies, and infrastructure of E-Commerce. |
| **CO3** | Examine security and legal frameworks applicable to E-Commerce. |
| **CO4** | Evaluate the ethical, social, and regulatory concerns in digital commerce. |
| **CO5** | Apply digital marketing strategies to online businesses. |
| **CO6** | Develop and demonstrate a basic E-Commerce prototype or campaign. |

**Course Contents:**

| **Unit** | **Description** | **Duration (Hrs.)** |
|---|---|---|
| 1. | **Introduction to E-Commerce:** Definition, history, and evolution; traditional vs. electronic commerce; E-Commerce framework and business impact; features and limitations of E-Commerce. | 5 Hrs |
| 2. | **E-Commerce Business Models:** B2B, B2C, C2C, C2B, G2C; revenue models; case studies (Amazon, Flipkart, Meesho, etc.); advantages and challenges. | 4 Hrs |
| 3. | **E-Commerce Infrastructure:** Internet and WWW, web hosting, domain registration, client-server architecture; overview of E-Commerce platforms (Shopify, Wix, WooCommerce). | 4 Hrs |

| | | |
|---|---|---|
| 4. | **Electronic Payment Systems & Security:** Payment methods: credit/debit cards, wallets, UPI, net banking; digital signatures; SSL; common threats and prevention; e-fraud case studies. | 5 Hrs |
| 5. | **Legal, Ethical & Regulatory Issues:** Overview of IT Act 2000, data protection laws (GDPR); privacy concerns; ethical and IPR issues in digital business. | 5 Hrs |
| 6. | **Digital Marketing & Strategy:** Introduction to SEO, SEM, affiliate marketing, email campaigns, and social media marketing; use of Google Ads, Meta Ads, CRM tools. | 5 Hrs |
| | **TOTAL** | 28 Hrs |

**Text Books:**

3. P.T. Joseph, "E-Commerce: An Indian Perspective", PHI Learning
4. Kenneth C. Laudon & Carol Guercio Traver, "E-Commerce 2023", Pearson

**Reference Books:**

- S.J. Joseph, "E-Commerce: A Managerial Perspective", Prentice Hall
- Elias M. Awad, "Electronic Commerce: From Vision to Fulfillment", Pearson

**E-Resources:**

- NPTEL: **https://nptel.ac.in**
- Google Digital Garage: **https://learndigital.withgoogle.com**
- HubSpot Academy: **https://academy.hubspot.com**

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| | |
|---|---|
| **Program:** B. Tech. (AI & ML ) | **Semester:** IV |
| **Course:** E-Commerce Lab. | **Code: AMMC402** |

| |
|---|
| **Guidelines for Instructor's Manual** |
| The instructor's manual should include course objectives, weekly practical plan, expected outcomes, and experiment details. It must define evaluation rubrics, provide links to online tools, and include sample case studies and project topics. |
| **Guidelines for Student's Lab Journal** |
| Students should document each practical with a title, objective, tools used, steps taken, and screenshots of implementation, observations, and conclusions. Each entry should be signed by the instructor weekly. |
| **Guidelines for Lab /TW Assessment** |
| Term work will be assessed based on attendance, timely submissions, completeness, understanding, and creativity in mini-projects. Evaluation should use standard rubrics assessing technical accuracy and communication of results. |
| **Guidelines for Practical Examination** |
| The practical examination should assess the student's coding ability, problem-solving skills, and conceptual understanding. It should consist of one programming problem from the performed experiments and an oral viva. Students must write and execute the solution independently within the given time frame. Evaluation should be based on execution correctness, code structure, and oral response, using standard rubrics. Both internal and external examiners must align on evaluation methods. |
| **Guidelines for Laboratory Conduction** |
| Laboratory sessions should begin with a brief explanation of the topic, objectives, and expected outcomes. Instructors should encourage active participation, individual coding practice, and peer learning. Faculty should monitor progress during the session and resolve student doubts. Weekly viva or discussion should be conducted for conceptual reinforcement. A lab timetable, attendance record, and assessment log should be maintained. The aim is to build consistent technical skills and analytical thinking through hands-on practice. |

**List of Assignments:-**

| | |
|---|---|
| 1. | Conduct a survey on online shopping behavior using Google Forms and analyze the results. |
| 2. | Create a basic business profile using Google My Business. |
| 3. | Design a sample product listing page using Canva or Figma. |
| 4. | Create a demo online store using Shopify, Wix, or WooCommerce (Trial version). |

| | |
|---|---|
| 5. | Simulate a digital payment flow using sandbox/test mode of Razorpay or Paytm. |
| 6. | Perform SEO analysis using Google Trends or Ubersuggest for a product. |
| 7. | Draft a privacy policy and terms of use for an e-commerce platform. |
| 8. | Create a mock digital marketing campaign using Meta Ads Manager or Canva. |
| 9. | Analyze and present a case study on any leading e-commerce platform (Amazon, Meesho, etc.). |
| 10. | Mini Project: Design a prototype of an online store with homepage, product section, and simulated cart. |

**Text Books:**

1. P.T. Joseph, "E-Commerce: An Indian Perspective", PHI Learning
2. Kenneth C. Laudon & Carol Guercio Traver, "E-Commerce 2023", Pearson

**Reference Books:**

- S.J. Joseph, "E-Commerce: A Managerial Perspective", Prentice Hall
- Elias M. Awad, "Electronic Commerce: From Vision to Fulfillment", Pearson

**E-Resources:**

- NPTEL: https://nptel.ac.in
- Google Digital Garage: https://learndigital.withgoogle.com
- HubSpot Academy: https://academy.hubspot.com

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** IV | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Quantitative Aptitude & Numerical Analysis | | | | | | | | **Code: AMAE402** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| - | 02 | - | 01 | - | . | 25 | - | - | 25 |

**Prerequisites:**

- Basic knowledge of Mathematics at Higher Secondary Level
- Familiarity with basic logical reasoning and problem-solving skills

**Course Objectives:**

- To develop and enhance quantitative aptitude and numerical problem-solving skills.
- To train students in basic numerical techniques useful for competitive exams and placements.
- To encourage analytical thinking and logical reasoning

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Solve basic and advanced quantitative aptitude problems. |
| **CO2** | Apply shortcut methods for fast calculations in aptitude questions. |
| **CO3** | Develop logical reasoning skills for competitive exams. |
| **CO4** | Use numerical techniques to solve real-life problems. |
| **CO5** | Improve accuracy and speed in solving numerical analysis problems. |
| **CO6** | Demonstrate confidence in attempting placement and entrance tests. |

**Guidelines for Instructor's Manual**

The instructor's manual should contain detailed solutions and clear explanations for each assignment included in the lab syllabus. It should also mention alternative methods for solving problems where applicable and provide additional practice questions to help students strengthen their understanding. The manual must support instructors in conducting the practical sessions smoothly and effectively.

**Guidelines for Student's Lab Journal**

Each student must maintain a well-organized lab journal, which should include the problem statement, a detailed step-by-step solution, and the final answer for every assignment. Students should write their observations and conclusions after completing each practical. The journal should reflect neatness, clarity, and completeness, ensuring it serves as a valuable reference for revision.

**Guidelines for Lab /TW Assessment**

Term work should be evaluated continuously throughout the semester. Marks should be awarded based on the regularity and punctuality of submissions, the correctness and completeness of solutions, and the

| |
|---|
| student's understanding of the methods used. Viva voce examinations should also be conducted periodically to assess the student's conceptual clarity and problem-solving approach. |
| **Guidelines for Practical Examination** |
| The practical examination should test the student's ability to solve given numerical and aptitude problems accurately within a limited time frame. Along with the problem-solving part, an oral examination should be conducted to check the student's understanding of the methods and logic applied during the practical's. The exam should ensure that students can confidently apply the techniques learned during the course. |
| **Guidelines for Laboratory Conduction** |
| Practical sessions should begin with an explanation of the relevant concepts and shortcut methods, followed by solving a few examples as a demonstration. Instructors should encourage students to discuss alternative approaches and share problem-solving tips among peers. Ample practice problems should be given during lab hours to reinforce learning and improve speed and accuracy. |
| **List of Assignments:-** |
| 1. **Experiment No. 1** |
| To solve problems based on number systems including even-odd numbers, divisibility rules, and digit sums using shortcut methods. |
| 2. **Experiment No. 2** |
| To find the Highest Common Factor (HCF) and Least Common Multiple (LCM) of numbers and apply shortcut techniques for fast calculations. |
| 3. **Experiment No. 3** |
| To solve various percentage-based problems related to profit, loss, discounts, and marked price efficiently using quick methods. |
| 4. **Experiment No. 4** |
| To apply profit and loss formulas to practical business scenarios and calculate gains or losses using fast techniques. |
| 5. **Experiment No. 5** |
| To solve problems on ratio, proportion, and partnership using direct and inverse methods and learn practical applications. |
| 6. **Experiment No. 6** |
| To analyze time and work problems involving efficiency, multiple workers, and pipes & cisterns using various shortcut approaches. |
| 7. **Experiment No. 7** |

| |
|---|
| To calculate time, speed, and distance for different situations including trains, boats & streams, and relative speed problems. |
| 8. **Experiment No. 8** |
| To compute simple and compound interest for different principal amounts, rates, and time periods and understand installment calculations. |
| 9. **Experiment No. 9** |
| To interpret and analyze data using tables, bar graphs, pie charts, and line graphs and solve related questions accurately. |
| 10. **Experiment No. 10** |
| To solve problems on permutations and combinations and apply counting principles for arrangements and selections. |
| 11. **Experiment No. 11** |
| To understand and solve basic probability problems including independent and dependent events and real-life applications. |
| 12. **Experiment No. 12** |
| To apply the properties of logarithms to simplify and solve exponential equations and real-life numerical problems. |
| **Text Books:** |
| 1. R.S. Aggarwal, *Quantitative Aptitude for Competitive Examinations*, S. Chand Publishing. |
| **Reference Books:** |
| • Arun Sharma, *How to Prepare for Quantitative Aptitude for CAT*, McGraw Hill.<br>• Abhijit Guha, *Quantitative Aptitude for Competitive Examinations*, Tata McGraw Hill. |
| **E-Resources:** |
| • www.indiabix.com<br>• www.lofoya.com<br>• YouTube Channels: *Unacademy*, *Gradeup* |

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | | **Semester:** IV |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Programming Lab – II | | | | | | | | | **Code: AMVS404** |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| - | 04 | - | 02 | - | - | 25 | - | 25 | 50 |

**Prerequisites:**

- Basic understanding of programming logic (preferably C/C++)
- Knowledge of flowcharts, algorithms, and data types

**Course Objectives:**

- To introduce Python programming language for general-purpose and application development.
- To develop the ability to write simple to intermediate Python programs.
- To implement object-oriented and modular programming in Python.
- To utilize Python libraries for real-world problems including data handling.

**Course Outcomes:** Upon successful completion of the course, students will be able to:

| | |
|---|---|
| **CO1** | Understand Python syntax, data types, and control structures. |
| **CO2** | Develop Python programs using functions, modules, and file handling. |
| **CO3** | Implement object-oriented programming concepts in Python. |
| **CO4** | Use libraries like NumPy and Pandas for data processing. |
| **CO5** | Solve real-world problems using Python scripting. |
| **CO6** | Analyze and debug Python programs to improve code efficiency and readability. |

**Course Contents:**

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| Sr. No | Description | Duration (Hrs.) |
|---|---|---|
| 1. | **Introduction to Python:** Overview of Python, features, applications, Python interpreter, comments, indentation, basic syntax, variables, keywords, operators, input/output, basic data types: int, float, string, Boolean, Python Casting | 4 Hrs |
| 2. | **Control Structures and Data Structures:** Conditional statements (if, elif, else), loops (for, while), loop control statements (break, continue). Data structures – lists, tuples, sets, dictionaries – creation, operations, and methods. | 4 Hrs |
| 3. | **Functions, Modules and Date-Time:** Defining and calling functions, argument types, return values, recursion. Lambda functions. Modules – built-in and user-defined. Packages and __main__,datetime, date, time, timedelta, Formatting dates using strftime / strptime | 4 Hrs |
| 4. | **Object Oriented Programming in Python:** Classes and objects, constructor, instance and class variables, inheritance, method overriding, polymorphism, encapsulation. | 4 Hrs |
| 5. | **File Handling, Exception Handling and Regular Expressions:** Opening, reading, writing files, file modes. Exception handling – try, except, else, finally, custom exceptions, re module, Searching, matching, replacing | 4 Hrs |
| 6. | **Data Processing with Libraries:** Introduction to NumPy – arrays, Dimensions in array, Indexing in Numpy, Numpy functions, vectorized operations. Pandas – Series, DataFrame, file I/O (CSV, Excel), basic operations. Simple data visualization using Matplotlib. | 4 Hrs |
| | **TOTAL** | 24 Hrs |

**Guidelines for Instructor's Manual**

The instructor's manual should briefly cover the university/program background, course objectives, syllabus, lab conduction methods, and assessment criteria. It should provide topic-wise concepts, expected outcomes, and a list of practical assignments using Python programming language. Sample code structures, algorithmic logic, and references should be included to support effective lab delivery. Standard evaluation rubrics, viva questions, and clarity on CO (Course Outcome) mapping must be provided. The manual is intended to ensure consistency in lab sessions and support instructors in aligning activities with the desired learning outcomes.

**Guidelines for Student's Lab Journal**

The student's lab journal should follow a uniform structure containing the experiment title, objective, problem statement, algorithm or pseudocode, program code, sample input/output, and conclusion. It should reflect individual work and understanding. Each journal must include a content index and a summary of weekly progress. Code documentation and neat presentation are expected. This record serves as the student's learning evidence and must be duly signed and verified each week by the lab instructor.

**Guidelines for Lab /TW Assessment**

The term work should be continuously assessed throughout the semester. Marks should be allotted based on timely submissions, code functionality, understanding of concepts, journal completion, and viva performance. Experiments should be graded using standardized rubrics focusing on logic design, correctness, output, and documentation. Bonus weightage may be given for case study analysis and the mini-project. The goal is to promote regular engagement, practical proficiency, and overall conceptual clarity.

**Guidelines for Practical Examination**

The practical examination should assess the student's coding ability, problem-solving skills, and conceptual understanding. It should consist of one programming problem from the performed experiments and an oral viva. Students must write and execute the solution independently within the given time frame. Evaluation should be based on execution correctness, code structure, and oral response, using standard rubrics. Both internal and external examiners must align on evaluation methods.

**Guidelines for Laboratory Conduction**

Laboratory sessions should begin with a brief explanation of the topic, objectives, and expected outcomes. Instructors should encourage active participation, individual coding practice, and peer learning. Faculty should monitor progress during the session and resolve student doubts. Weekly viva or discussion should be conducted for conceptual reinforcement. A lab timetable, attendance record, and assessment log should be maintained. The aim is to build consistent technical skills and analytical thinking through hands-on practice.

**List of Assignments:- (Perform Any 10 Practical's)**

1. **Conditional and Looping Statements**

Write a Python program to accept a number from the user and check whether it is **even or odd**.
Then, use a for loop to display the multiplication table of that number from 1 to 10.

| |
|---|
| 2. **List, Tuple, Dictionary, and Set Operations** |
| Create a list of five student names and display each using a loop.<br>Then create a dictionary to store the names and their marks, and display the highest marks using a dictionary method. |
| 3. **User-defined Functions, Arguments, Recursion** |
| Write a function to calculate the **simple interest** using parameters: principal, rate, and time.<br>Also write a recursive function to find the **factorial** of a number entered by the user. |
| 4. **Python Modules, Importing Techniques and Date-Time** |
| Create a user-defined module math_utils.py with functions for square() and cube().<br>Write a main Python program that imports this module and uses these functions for user input.<br>Also, ask the user for their birthdate (in DD-MM-YYYY format) and display:<br>● Their age in years<br>● Day of the week they were born<br>● Days left until their next birthday |
| 5. **Class Definition, Object Creation, Inheritance** |
| Create a class Student with attributes name and roll number.<br>Add a method to display student details. Create an object of the class and call the method. |
| 6. **Exception Handling with File I/O** |
| Write a program that accepts a string from the user and writes it into a file data.txt.<br>Then, read the content back and display it. Handle exceptions like file not found and input errors. |
| 7. **Pattern Matching using Regular Expressions** |
| Write a Python program to:<br>● Validate an email address using regular expressions<br>● Extract all mobile numbers from a given multiline string<br>● Replace all whitespace with a hyphen in a given sentence |
| 8. **NumPy Arrays and Operations** |
| Create a NumPy array of 5 elements and display the array.<br>Perform operations like array addition, mean, and sorting. |
| 9. **Pandas DataFrames and CSV File Operations** |

Create a DataFrame from a dictionary containing employee names and salaries.
Display the DataFrame and calculate the average salary using Pandas.

10. **Data Visualization using Matplotlib**

Write a Python program using Matplotlib to draw a **bar chart** showing the marks of 5 subjects.
Add proper labels and title.

11. **Case Study: Simple Automation Tool using Python**

Write a program that accepts a list of filenames from the user and checks which of them exist in the current directory using the os module.

12. **Mini Project: Integrated Application using Multiple Concepts**

Create a simple **student record keeper** that stores name and marks using a dictionary.
Allow the user to add, search, and delete records using functions.

**Text Books:**

1. Reema Thareja, Python Programming, Oxford University Press
2. E. Balagurusamy, Introduction to Computing and Problem Solving with Python, McGraw-Hill

**Reference Books:**

- Mark Lutz, Learning Python, O'Reilly
- Allen B. Downey, Think Python, O'Reilly

**E-Resources:**

- NPTEL Online Courses: https://nptel.ac.in
- W3Schools: https://www.w3schools.com
- GeeksforGeeks: https://www.geeksforgeeks.org
- Coursera: https://www.coursera.org

## DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

| **Program:** B. Tech. (AI & ML ) | | | | | | | | **Semester:** IV | |
|---|---|---|---|---|---|---|---|---|---|
| **Course:** Internship-III | | | | | | | | **Code: AMIN403** | |
| **Teaching Scheme** | | | | **Evaluation Scheme** | | | | | |
| **Lecture** | **Practical** | **Tutorial** | **Credit** | **CIE** | **ETE** | **TW** | **OR** | **PR** | **Total** |
| 4 Week | | | 02 | | - | 25 | - | - | 25 |
| | | | | | | | | | |
| Kindly refer the policy of internship on College Website | | | | | | | | | |