Kurukshetra University, Kurukshetra

(Established by the State Legislature Act-XII of 1956)
(“A++” Grade, NAAC Accredited)

Syllabus
for

Post Graduate Programme

Master of Computer Applications

as per NEP-2020
Curriculum and Credit Framework for Postgraduate Programme

With Multiple Entry-Exit, Internship and CBCS-LOCF
With effect from the session 2024-25 (in phased manner)

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
FACULTY OF SCIENCES

KURUKSHETRA UNIVERSITY, KURUKSHETRA -136119

CC-1 Client-side Web Technology

With effect from the Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	1st
Name of the Course	Client-side Web Technology
Course Code	M24-CAP-101
Course Type	CC-1
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course aims to provide a comprehensive understanding of front-end development using the MERN stack, covering HTML, CSS, and JavaScript basics. Students will learn about React for building dynamic user interfaces, including components, state management, and event handling. The course also explores advanced topics such as React Router, Redux for state management, and advanced hooks for managing side effects and context.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Gain an understanding of the web development process and the components of the MERN stack, with a focus on HTML structure, CSS styling, and responsive design. CLO-2 Develop foundational JavaScript skills, including control structures, functions, objects, arrays, and DOM manipulation for dynamic web interactions. CLO-3 Learn the basics of React, including JSX, components, state management, lifecycle methods, and handling events and forms within React applications. CLO-4 Master advanced React topics like React Router for navigation, state management with Redux, and using advanced hooks for managing complex state and side effects.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Basics of Front End Development: Overview of web development (Front End vs. Back End), Understanding the MERN stack and its components, Tools and environments (text editors, browsers, version control with Git); HTML (HyperText Markup Language): Structure of an HTML document, HTML elements and attributes, Forms and input types, Semantic HTML (header, footer, article, section, nav); CSS (Cascading Style Sheets): Basics of CSS (syntax, selectors, properties), CSS Box Model, Positioning and layout (float, flexbox, grid), Responsive design (media queries, mobile-first design).	15
II	Basics of JavaScript: Introduction to JavaScript, Variables, data types, and operators, Control structures (if, else, switch, loops); Functions and Scope: Defining and invoking functions, Function expressions and arrow functions, Scope and closures; Objects and Arrays: Creating and manipulating objects, Array methods and iteration; Regular Expressions: Introduction to RegExp, Regular expression usage, Modifiers, RegExp	15

	patterns, RegExp methods, String methods for RegExp; DOM Manipulation and Events: Selecting and manipulating DOM elements, Event handling and delegation, Creating and appending elements dynamically	
III	Introduction to React: Overview and advantages of React, Setting up a React development environment (using Create React App); JSX (JavaScript XML): Understanding JSX syntax, Embedding expressions in JS, JSX best practices; Components and Props: Functional and class components, Props and component communication, Prop types and default props.; State and Lifecycle: Understanding state in React, State management in class components, Lifecycle methods (componentDidMount, componentDidUpdate, componentWillUnmount); Event Handling and Forms: Handling events in React, Controlled vs. uncontrolled components, Form handling and validation	15
IV	React Router: Introduction to React Router, Setting up and configuring routes, Navigating between routes and passing parameters; State Management with Redux: Introduction to Redux, Setting up Redux with React, Actions, reducers, and store, Connecting Redux to React components; Advanced Hooks: Using built-in hooks (useEffect, useContext, useReducer), Creating custom hooks, Managing side effects with useEffect	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media.
2) Kogent Learning. (2009). *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book*. Wiley India Pvt. Ltd.
3) Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*. Wiley.
4) Robson, E., & Freeman, E. (2014). *Head First JavaScript Programming: A Brain-Friendly Guide*. O'Reilly Media.
5) Banks, A., & Chinnathambi, K. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.

CC-2 Operating System and Linux

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	1st		
Name of the Course	Operating System and Linux		
Course Code	M24-CAP-102		
Course Type	CC-2		
Level of the course (As per Annexure-I	400-499		
Pre-requisite for the course (if any)	-		
Course Objectives	This course provides a foundational understanding of operating systems, covering their definition, types, and functions. Students will explore system structures, process management, CPU scheduling, memory management, paging and segmentation, virtual memory, and file systems. Additionally, the course offers an introduction to Linux, including its history, architecture, file system, basic commands, shell scripting, process and user management, networking, system administration, and basic security concepts.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Understand the fundamental concepts, functions, and structures of operating systems, and apply various CPU scheduling algorithms. CLO-2 Grasp memory hierarchy, allocation techniques, paging, segmentation, virtual memory concepts, and file system management. CLO-3 Learn the history, features, and architecture of Linux, perform basic file operations, and write simple shell scripts. CLO-4 Manage processes, users, and groups in Linux, utilize network commands, perform system administration tasks, and understand basic security measures.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		
Part B- Contents of the Course			

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Introduction to Operating Systems: Definition, types, and functions of an operating system; System Structures: Operating system services, system calls, system programs, and system structure; Process Management: Process concept, process scheduling, operations on processes, inter-process communication; CPU Scheduling: Scheduling criteria, scheduling algorithms (FCFS, SJF, Priority, Round Robin, Multilevel Queue Scheduling).	15
II	Memory Management: Memory Hierarchy, Types of memory, memory allocation techniques; Paging and Segmentation: Basic concepts, paging, segmentation, segmentation with paging; Virtual Memory: Demand paging, page replacement algorithms, allocation of frames, thrashing; File Systems: File concepts, access methods, directory and disk structure, file system mounting, file sharing, protection.	15
III	Introduction to Linux: History, features, architecture of Linux; Linux File System: File and directory structure, file permissions, standard file types; Basic Commands: File and	15

	directory operations (ls, cp, mv, rm, mkdir), text processing (cat, grep, sort), system status (ps, top, df, du); Shell Scripting: Introduction to shell, shell variables, control structures (if, case, while, for), writing simple shell scripts.	
IV	Process Management in Linux: Managing processes (ps, top, kill, nice), job scheduling (cron, at); User and Group Management: Creating and managing users and groups, file permissions, changing ownership (chown, chgrp); Networking in Linux: Basic network commands (ifconfig, ping, netstat, ssh), configuring network interfaces; System Administration: Package management (installing and removing software using rpm, dpkg, apt-get), backup and restore, logging; Security: Basic security concepts, user authentication.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
1) Class Participation:	5	Written Examination	
2) Seminar/presentation/assignment/quiz/class test etc.:	10		
3) Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
2) Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.
3) Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
4) Love, R. (2013). Linux System Programming (2nd ed.). O'Reilly Media.
5) Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook (5th ed.). Pearson.
6) Sobell, M. G. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming (4th ed.). Pearson.
7) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide (3rd ed.). McGraw-Hill Education.
8) Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press.

CC-3 Data Structures

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	1^{st}
Name of the Course	Data Structures
Course Code	M24-CAP-103
Course Type	CC-3
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course introduces fundamental concepts of algorithms and data structures, including algorithmic notation, programming principles, and program analysis. Students will explore arrays, searching and sorting techniques, stacks, queues, and linked lists, along with their applications. The course also covers tree structures such as binary trees, AVL trees, B-trees, and tries, as well as graph terminology, representation, and traversal methods. Additionally, students will learn about set operations, file queries, sequential organization, index techniques, and external sorting.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Master algorithmic notation, programming principles, and implement arrays, searching and sorting techniques. CLO-2 Apply stack and queue operations, understand linked lists, and their applications including dynamic storage management. CLO-3 Comprehend binary trees, binary search trees, AVL trees, B-trees, B+ tree indexing, Trie tree indexing, and their applications. CLO-4 Utilize graph representations, traversals, applications, sets operations, and file organization techniques.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction: Algorithmic notation – Programming principles – Creating programs-Analyzing programs. Arrays: One dimensional array, multidimensional array, pointer arrays. Searching: Linear search, Binary Search, Fibonacci search. Sorting techniques: Internal sorting - Insertion Sort, Selection Sort, Shell Sort, Bubble Sort, Quick Sort, Heap Sort, Merge Sort and Radix Sort.	15
II	Stacks: Definition – operations - applications of stack. Queues: Definition - operations - Priority queues – Dequeues – Applications of queue. Linked List: Singly Linked List, Doubly Linked List, Circular Linked List, linked stacks, Linked queues, Applications of Linked List – Dynamic storage management – Generalized list.	15
III	Trees: Binary tree, Terminology, Representation, Traversals, Applications – Binary search tree – AVL tree. B Trees: B Tree indexing, operations on a B Tree, Lower and upper bounds of a B Tree - B + Tree Indexing – Trie Tree Indexing.	15
IV	Graph: Terminology, Representation, Traversals – Applications - spanning trees, shortest path and Transitive closure, Topological sort. Sets: Representation - Operations on sets – Applications. Files: queries - Sequential organization – Index techniques. External sorting.	15

		Total Contact Hours	60
Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Horowitz, E., & Sahni, S. (2004). *Fundamentals of Data Structures*. Galgotia Book Source Pvt. Ltd.
2) Samanta, D. (2012). *Classic Data Structures* (2nd ed.). Prentice-Hall of India Pvt. Ltd., India.
3) Kruse, R., Tondo, C. L., & Leung, B. (2007). *Data Structures and Program Design in C* (2nd ed.). Prentice-Hall of India Pvt. Ltd.
4) Weiss, M. A. (2006). *Data Structures and Algorithm Analysis in C* (2nd ed.). Pearson Education.

CC-4 Programming in JAVA

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	1st
Name of the Course	Programming in JAVA
Course Code	M24-CAP-104
Course Type	CC-4
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a comprehensive introduction to Java, covering its history, features, and applications. Students will learn Java programming basics, including syntax, variables, control flow, methods, and arrays. The course also delves into object-oriented programming concepts such as classes, objects, encapsulation, inheritance, polymorphism, and interfaces. Additionally, students will explore advanced topics like exception handling, file handling, multithreading, event handling, generics, JDBC for database connectivity, and GUI programming with Swing.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Understand Java's background, features, and apply fundamental programming concepts including variables, operators, control flow, methods, and arrays. CLO-2 Master object-oriented programming principles including classes, objects, inheritance, polymorphism, interfaces, and packaging in Java. CLO-3 Gain proficiency in handling exceptions, working with files, implementing multithreading, and utilizing Java Collections for efficient data management. CLO-4 Explore and utilize advanced Java features such as generics, lambda expressions, JDBC for database connectivity, and GUI programming with JavaFX or Swing.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Java: History, features, and applications; Basics of Java programming: Syntax, variables, data types, operators, expressions, and statements; Control flow: Decision-making statements (if, else-if, switch), looping statements (for, while, do-while), and branching; Methods: Declaring methods, passing parameters, method overloading, and recursion; Arrays: Declaring, initializing, and manipulating arrays. Array operations and algorithms.	15
II	Classes and Objects: Declaring classes, creating objects, constructors, and instance variables; Encapsulation: Access modifiers (public, private, protected, default), getters, and setters; Inheritance: Extending classes, method overriding, super keyword, and method overloading; Polymorphism: Method overriding, dynamic method dispatch, and abstract classes; Interfaces: Defining interfaces, implementing interfaces, and using interface	15

	references; Packages: Creating and using packages, importing classes and packages.	
III	Exception Handling: Understanding exceptions, try-catch block, throw and throws keywords, and finally block; File Handling: Reading from and writing to files using FileInputStream, FileOutputStream, FileReader, and FileWriter; Multithreading: Creating threads, thread lifecycle, synchronization, thread communication. Applet programming, Applet life Cycle, Applet Graphics programming.	15
IV	Event Handling: AWT Classes, ActionListener, MouseListener, MouseMotionListener, Layout managers, Generics: Introduction to generics, generic classes and generic methods, Java Database Connectivity (JDBC): Connecting to databases, executing SQL queries, handling transactions, and managing resources; GUI Programming: Introduction to Swing for creating graphical user interfaces (GUIs).	15
Total Contact Hours		60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Balaguruswamy, E. (2009). *Programming with JAVA: A Primer*. Tata McGraw Hill.
2) Naughton, P., & Schildt, H. (2002). *The Complete Reference Java 2*. Tata McGraw Hill.
3) Neimeyer, P., & Peck, J. (1996). *Exploring Java*. O'Reilly.
4) Hahn, H. (1996). *Teach Yourself the Internet*. Prentice-Hall of India (P.H.I.).
5) Boone, B., & Stanek, W. (2001). *Java 2 Exam Guide*. Tata McGraw Hill.

PC-1 PRACTICAL-1

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-1
Course Code	M24-CAP-105
Course Type	PC-1
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This is a laboratory course and the objective of this course is to acquaint the students with the understanding and implementing of client-side web technologies. Also, the concepts of operating systems and shell programming will be implemented by the students.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO 1: Solve practical problems related to theory courses undertaken in the CC-1 and CC-2 from application point of view. CLO 2: Know how to use the client-side web technologies. CLO 3: implement the various functions of operating systems. CLO 4: Designing and implementing the shell programs in Linux.

Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	Contact Hours
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 questions from the Part-B.	120
Part-A **HTML/CSS Basics:** • Creating a webpage structure with HTML. • Styling the webpage using CSS (inline, internal, and external styles). **Responsive Design:** • Making the webpage responsive using media queries. • Using frameworks like Bootstrap for responsive design. **JavaScript Basics:** • Adding interactivity with JavaScript (DOM manipulation, event handling). • Working with variables, loops, and conditions. **Frameworks and Libraries:** • Using front-end frameworks React. • Utilizing libraries such as jQuery for DOM manipulation. **Introduction to React:** • Create a simple React component that displays "Hello, World!" on the screen. • Use JSX syntax and explain its advantages over plain JavaScript. **State and Props:** • Build a component that takes props and renders them. • Implement state in a component and update it based on user interaction (e.g., button click). **Basic Todo App:** Develop a Todo application where users can add, delete, and mark tasks as completed. Use state to manage the list of tasks. **Using React Router:**	60

- Set up React Router in a project and create multiple pages (e.g., Home, About, Contact).
- Implement navigation between these pages using Link and NavLink.

Redux Integration:

- Integrate Redux for state management in a React application.
- Implement actions, reducers, and connect components to Redux store.

Responsive Design with React Router:

- Build a responsive multi-page application using React Router.
- Ensure layout adjustments for different screen sizes using CSS media queries or frameworks like Bootstrap.

Part-B

60

(Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

1) Implement a simple program demonstrating the creation and synchronization of threads or processes.
2) Design and simulate a memory management system (e.g., paging, segmentation).
3) Implement algorithms like First Fit, Best Fit, and Worst Fit for memory allocation.
4) Implement a basic file system with operations like file creation, deletion, reading, and writing.
5) Compare different file allocation methods (e.g., contiguous allocation, linked allocation, indexed allocation).
6) Solve synchronization problems such as the producer-consumer problem or dining philosophers problem using semaphores or mutexes.
7) Implement a solution for deadlock prevention, avoidance, or detection.
8) Profile and analyze the performance of different scheduling algorithms (e.g., FCFS, SJF, Round Robin) using simulations.
9) Evaluate the impact of caching and paging strategies on system performance.
10) Write a shell script named hello.sh that prints "Hello, World!" to the terminal when executed.
11) Demonstrate running the script and explain how to make it executable using chmod.
12) Write a script greet_user.sh that prompts the user for their name and then prints a personalized greeting.
13) Use variables to store user input and demonstrate the use of read command.
14) Create a script check_number.sh that accepts a number as an argument.
15) Check if the number is positive, negative, or zero, and print an appropriate message using conditional statements (if-else).
16) Develop a script countdown.sh that takes a number as input and prints a countdown from that number to 1.
17) Use a loop (e.g., while or for) to implement the countdown.
18) Write a script file_info.sh that accepts a filename as an argument.
19) Check if the file exists and whether it is a regular file or directory. Display appropriate messages based on the checks.
20) Create a script word_count.sh that reads a text file (provided as an argument) and counts the number of words in the file.
21) Utilize command-line tools like wc and cat for reading and counting words.

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media.
2) Kogent Learning. (2009). *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book*. Wiley India Pvt. Ltd.
3) Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*. Wiley.
4) Robson, E., & Freeman, E. (2014). *Head First JavaScript Programming: A Brain-Friendly Guide*. O'Reilly Media.

5) Banks, A., & Chinnathambi, K. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.

6) Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.

7) Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.

8) Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.

9) Love, R. (2013). Linux System Programming (2nd ed.). O'Reilly Media.

10) Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook (5th ed.). Pearson.

11) Sobell, M. G. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming (4th ed.). Pearson.

12) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide (3rd ed.). McGraw-Hill Education.

13) Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press

PC-2 PRACTICAL-2

With effect from Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-2
Course Code	M24-CAP-106
Course Type	PC-2
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This is a laboratory course and the objective of this course is to acquaint the students with the understanding and implementation of various data structures. Also, the students will implement the concepts of programming with Java.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO 1: Solve practical problems related to theory courses undertaken in the CC-3 and CC-4 from an application point of view. CLO 2: Know how to use and implement the various data structures. CLO 3: Implement the various features of Java Programming by writing suitable programs. CLO 4: Designing and implementing applications in Java.

Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course	
Practicals	**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 questions from the Part-B.	120
Part-A **Task 1: Linked List Implementation** • Implement a singly linked list in a programming language of your choice (e.g., C/C++, Java, Python). • Include functions/methods for insertion (at the beginning, end, and specific position), deletion, and traversal. **Task 2: Stack Operations** • Implement a stack using an array or linked list. • Include functions/methods for push, pop, peek, and checking if the stack is empty or full. **Task 3: Queue Implementation** • Implement a queue using an array or linked list. • Include functions/methods for enqueue, dequeue, peek, and checking if the queue is empty or full. **Task 4: Binary Search Tree (BST) Operations** • Implement a binary search tree (BST) in your chosen programming language. • Include functions/methods for insertion, deletion, searching for a key, finding minimum and maximum values, and traversing the tree (inorder, preorder, postorder). **Task 6: Sorting Algorithms** • Implement at least two sorting and searching algorithms (e.g., selection sort, insertion sort, merge sort, quick sort). • Compare their time complexity and performance using different input sizes.	60

Task 7: Graph Representation and Algorithms

- Implement an adjacency list representation of a graph.
- Include functions/methods for BFS (Breadth-First Search) and DFS (Depth-First Search) traversal of the graph.

Part-B

60
(Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

1) Write a Java program that converts temperatures between Celsius and Fahrenheit based on user input using methods for conversion and input validation.
2) Implement a Java program to perform matrix addition, multiplication, and transpose operations using arrays and methods.
3) Develop a Java program that converts a decimal number to its binary, octal, and hexadecimal equivalents using loops and methods.
4) Create a Java program to simulate a simple bank account management system with features like deposit, withdrawal, and balance inquiry using classes, objects, and encapsulation.
5) Write a Java program that reads a text file, counts the occurrences of each word, and displays the top N most frequent words using HashMap for storage and sorting.
6) Implement a Java program to generate the first N prime numbers using a combination of loops, methods, and optimizations like the Sieve of Eratosthenes algorithm.
7) Develop a Java program that takes a month and year as input and prints the calendar for that month using control flow statements and loops for date calculation.
8) Write a Java program that generates different number patterns like pyramid patterns using nested loops and methods for pattern printing.
9) Create a Java program to manage an employee payroll system with features for adding employees, calculating salaries based on hours worked or monthly salary, and generating pay slips using classes, inheritance, and polymorphism.
10) Implement Java programs to compare the performance of different sorting algorithms (like quicksort, mergesort, and heapsort) on large arrays of integers, measuring and analyzing time complexity.
11) Develop a Java program that recursively searches a directory for files matching a given pattern and displays the file paths using recursion and file handling classes.
12) Write a Java program to perform arithmetic operations (addition, subtraction, multiplication, division) on large numbers using BigInteger class and exception handling for division by zero.
13) Implement a Java program to solve the Tower of Hanoi problem for N disks using recursion, demonstrating the steps and movements required.
14) Write a Java program to find the largest and smallest elements in an array.
15) Implement a Java program to sort an array of integers using bubble sort.
16) Create a Java program to find the frequency of each element in an array.
17) Develop a Java program to reverse an array without using an additional array.
18) Write a Java program to merge two sorted arrays into a single sorted array.
19) Define a Java class representing a Student with private instance variables and public getter and setter methods.
20) Create a Java program to demonstrate constructor overloading in a class.
21) Implement a Java program to calculate the area and perimeter of a rectangle using a class and object.
22) Develop a Java program to implement inheritance by creating a base class Animal and derived classes like Dog and Cat.
23) Write a Java program to demonstrate method overriding by implementing a base class Shape and derived classes like Circle and Rectangle.

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Horowitz, E., & Sahni, S. (2004). *Fundamentals of Data Structures*. Galgotia Book Source Pvt. Ltd.
2) Samanta, D. (2012). *Classic Data Structures* (2nd ed.). Prentice-Hall of India Pvt. Ltd., India.
3) Kruse, R., Tondo, C. L., & Leung, B. (2007). *Data Structures and Program Design in C* (2nd ed.). Prentice-Hall of India Pvt. Ltd.
4) Weiss, M. A. (2006). *Data Structures and Algorithm Analysis in C* (2nd ed.). Pearson Education.
5) Balaguruswamy, E. (2009). *Programming with JAVA: A Primer*. Tata McGraw Hill.
6) Naughton, P., & Schildt, H. (2002). *The Complete Reference Java 2*. Tata McGraw Hill.
7) Neimeyer, P., & Peck, J. (1996). *Exploring Java*. O'Reilly.
8) Hahn, H. (1996). *Teach Yourself the Internet*. Prentice-Hall of India (P.H.I.).
9) Boone, B., & Stanek, W. (2001). *Java 2 Exam Guide*. Tata McGraw Hill.

BC-1 Computer Fundamentals and Problem Solving Through C

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	1st
Name of the Course	Computer Fundamentals and Problem Solving Through C
Course Code	M24-CAP-108
Course Type	BC-1
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	The objective of this course is to provide a foundational understanding of computer systems, including hardware and software components, and to introduce essential concepts of digital systems, number systems, and Boolean logic. The course also aims to develop proficiency in programming using the C language, focusing on control structures, functions, data structures, and pointers. By the end of the course, students will be able to apply fundamental programming techniques to solve computational problems and have a strong grasp of the underlying principles of digital logic and computing.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Students will be able to explain the basic organization of a computer and understand the purpose and methods of program planning using algorithms, flowcharts, and pseudocodes. CLO-2. Students will develop the ability to represent and manipulate information using various number systems, binary arithmetic, and Boolean logic. CLO-3. Students will acquire proficiency in programming with the C language, including the use of data types, operators, control structures, and input/output operations. CLO-4. Students will demonstrate the ability to create modular programs in C using functions, effectively manage data structures such as arrays, strings, and files, and work with pointers to manipulate memory and data efficiently.

Credits	Theory	Practical	Total
	0	0	0
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Computer Fundamentals: Basics of computers, basic computer organization, storage hierarchy, storage devices, input-output devices. Computer Software. Introduction to operating systems. Planning the computer program: Purpose of program planning, algorithm, flowcharts, decision tables, pseudocodes.	15
II	Digital Fundamentals: Information representation - number systems, number system conversion; Computer codes - BCD code, EBCDIC code, ASCII, Unicode; Binary arithmetic; Binary logic - Boolean algebra, Boolean functions, truth table, simplification of	15

	Boolean functions (upto 4 variables only), K-map, digital logic gates.	
III	Elements of C language: C character set, identifiers & keywords, data types: declaration & definition. Operators: Arithmetic relational, logical, bitwise, unary, assignment and conditional operators & their hierarchy & associativity, Data input/output. Control statements: Sequencing, Selection: if and switch statement; iteration: for, while, and do-while loop; break, continue, goto statement.	15
IV	Functions in C language: Definition, prototype, passing parameters, recursion, Data structure: arrays, structures, union, string, data files. Pointers: Declaration, operations on pointers, array of pointers, pointers to arrays.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

Balagurusamy, E. *Programming in ANSI C*. 8th ed., McGraw Hill, 2019. ISBN: 9789353165129.

Morris Mano, M. *Digital Logic and Computer Design*. 1st ed., Pearson, 2016. ISBN: 9789332551763.

Forouzan, Behrouz A. *Fundamentals of Computer Science: Computer Essentials*. 3rd ed., Cengage Learning, 2008. ISBN: 9788131512456.

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*. 2nd ed., Pearson Education, 1988. ISBN: 9780131103627.

BC-2 PRACTICAL-3

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-3
Course Code	M24-CAP-109
Course Type	BC-2
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This course focuses on hands-on experience with computer fundamentals. They will engage in program planning by creating and testing algorithms, flowcharts, and pseudocodes. Practical sessions will deepen their understanding of digital fundamentals through exercises on number systems, Boolean logic, and binary arithmetic. The course will provide extensive practice in C programming, allowing students to implement various data types, control structures, functions, and pointers in real-world coding tasks.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1: Design and implement efficient algorithms using flowcharts, pseudocodes, and decision tables to solve complex problems. CLO-2: Write C programs that demonstrate a strong understanding of control structures, data types, and operators to create optimized solutions. CLO-3: Develop modular C programs using functions, effectively managing code complexity and promoting reusability. CLO-4: Utilize pointers and data structures in C to enhance program efficiency and handle dynamic memory management in real-world applications.

Credits	Theory	Practical	Total
	0	0	0
Teaching Hours per week	0	4	4
Internal Assessment Marks	0	15	15
End Term Exam Marks	0	35	35
Max. Marks	0	50	50
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	**Contact Hours**
The examiner will set 3 questions at the time of practical examination by taking course learning outcomes (CLO) into consideration. The examinee will be required to write and execute 2 questions.	60
1) Implement a program using the conditional (ternary) operator to find the largest of three numbers. 2) Create a C program that acts as a simple calculator, performing addition, subtraction, multiplication, or division based on user input using the switch statement. 3) Write a program that uses `if-else` statements to determine whether a given year is a leap year or not. 4) Develop a C program using a `for` loop to print the multiplication table of a given number up to 10. 5) Write a C program to calculate the factorial of a number using both `while` and `do-while` loops. 6) Implement a program that uses `break` and `continue` statements within a loop to skip printing even numbers and stop the loop if the number exceeds 50. 7) Write a C program with a function that takes an integer as input and returns the	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

square of the number. Call this function from `main()`.

8) Develop a program that includes a function to calculate the area of a circle given the radius. Use `float` as the return type.
9) Create a C program that calculates the nth Fibonacci number using recursion.
10) Write a program that uses a function to find the maximum value in an integer array. The array should be passed to the function as a parameter.
11) Implement a program that uses functions to reverse a string and check if the string is a palindrome.
12) Write a C program that defines a structure to store student details (name, roll number, marks in three subjects) and calculates the total and average marks. Use a union to demonstrate memory sharing between different types.
13) Pointers in C Language
14) Pointer Basics: Write a program that demonstrates the use of pointers by printing the address and value of a variable using both the variable itself and a pointer to the variable.
15) Create a C program to store an array of strings (names of students) using an array of pointers. Display the names in reverse order.
16) Implement a program that uses a pointer to a function to pass a function as a parameter to another function, e.g., passing a function that calculates the square of a number to another function that prints it.

Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Practicum**	**15**	➢ **Practicum**	**35**
• Class Participation:	4	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	4		
• Mid-Term Examination:	7		

Part C-Learning Resources

Reference Books:

1) Balagurusamy, E. *Programming in ANSI C*. 8th ed., McGraw Hill, 2019. ISBN: 9789353165129.
2) Morris Mano, M. *Digital Logic and Computer Design*. 1st ed., Pearson, 2016. ISBN: 9789332551763.
3) Forouzan, Behrouz A. *Fundamentals of Computer Science: Computer Essentials*. 3rd ed., Cengage Learning, 2008. ISBN: 9788131512456.
4) Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*. 2nd ed., Pearson Education, 1988. ISBN: 9780131103627.

CC-5 Server Side Web Technology

With effect from the Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	2nd
Name of the Course	Server Side Web Technology
Course Code	M24-CAP-201
Course Type	CC-5
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides an in-depth understanding of web servers, client-server architecture, and Node.js, focusing on non-blocking I/O, event-driven programming, and package management. Students will learn to handle files, build HTTP servers, and manage asynchronous tasks while mastering error handling and debugging. With Express.js, they will design RESTful APIs, implement routing and middleware, and integrate user authentication. The course also introduces MongoDB for NoSQL database operations, including CRUD and indexing, equipping students to build scalable, secure web applications.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will be able to set up a Node.js environment, understand its non-blocking I/O and event-driven architecture, and manage packages and modules effectively. **CLO-2:** Students will gain the ability to handle files and directories, create robust HTTP servers, and implement event-driven programming while managing asynchronous tasks and debugging errors. **CLO-3:** Students will be able to develop Express.js applications with structured routing, middleware, and RESTful APIs, including secure user authentication using JWT and sessions. **CLO-4:** Students will learn to set up and manage MongoDB databases, perform CRUD operations, and utilize indexes to optimize query performance for NoSQL applications.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to web servers, Client-Server Architecture, Request-Response Cycle, Server-Side vs. Client-Side. Introduction to Node.js: Overview of Node.js, Non-blocking I/O, Event-driven architecture, Benefits of using Node.js in the MERN stack. Installing Node.js, Using Node Package Manager (npm), Creating and managing packages. Modules: Working with core modules, Creating and importing custom modules, require and exports.	15
II	File handling: Reading from and writing to files, Handling directories, Managing asynchronous tasks efficiently. Building Web Servers: Creating a basic HTTP server, Handling HTTP requests and	15

	responses, Understanding request methods (GET, POST, PUT, DELETE). Event-Driven Programming Using EventEmitter, Creating custom events, Handling real-time data. Error Handling and Debugging: Try-catch blocks, Handling asynchronous errors, Using debugging tools (e.g., node --inspect, Chrome DevTools).	
III	Express.js Basics: Introduction to Express.js, Setting up Express projects, Understanding routing and middleware. Using template engines (e.g., EJS) for server-side rendering, Designing RESTful APIs, CRUD operations, Structuring API routes. Built-in middleware (e.g., body-parser), Creating custom middleware, Error handling middleware. User authentication using JWT (JSON Web Tokens) and sessions.	15
IV	Introduction to MongoDB: NoSQL vs. SQL databases, Setting up MongoDB locally and on cloud (e.g., MongoDB Atlas), Document-based NoSQL database, JSON-like documents. Setting Up MongoDB: Installation, creating databases, collections, and documents CRUD Operations in MongoDB: Inserting, querying, updating, deleting documents Indexes in MongoDB: Creating and using indexes	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) "Node.js Design Patterns" by Mario Casciaro and Luciano Mammino
2) "Learning Node.js Development" by Andrew Mead
3) "Express in Action" by Evan M. Hahn
4) "REST API Development with Node.js" by Fernando Doglio
5) "MongoDB: The Definitive Guide" by Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow
6) "Learning MongoDB" by Amit Phaltankar, Juned Ahsan, and Michael Harrison

CC-6 Computer Network

With effect from the Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	2nd
Name of the Course	Computer Network
Course Code	M24-CAP-202
Course Type	CC-6
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	The course aims to provide a comprehensive understanding of network characterization, design issues, and service models, focusing on the OSI and TCP/IP reference models and their practical applications. It covers data communication concepts, including performance parameters, transmission media, modulation techniques, and switching methods, emphasizing the role of wired and wireless networks. The course delves into the data link layer, exploring protocols, error detection, media access, and IEEE standards, alongside advancements in wireless technologies like Wi-Fi, Wi-Max, and Bluetooth. It further examines the transport and network layers, addressing routing algorithms, congestion control, and QoS mechanisms, with a detailed focus on IPv4, IPv6, and protocols like TCP and UDP.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** characterize various types of computer networks and standards along with an insight into the principles of networking by using protocol layering of the Internet and the TCP/IP protocol suite. **CLO-2:** comprehend the notion of data communication and its related functional components and aspects. **CLO-3:** understand design issues related to Local area Networks and get acquainted with the prevailing wired and wireless LAN technology standards. **CLO-4:** get versed with the routing, addressing, congestion control, and security issues in Networks and the Internet architecture .

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Network Characterization**: Goals and Applications; Categorization according to Size, Purpose, Design issues & Transmission Technologies; Network Architecture and Service Models; Design issues for the Layers; Reference Models: OSI and TCP/IP; Functions of layers and protocols of TCP/IP; Comparison of OSI & TCP/IP ; Data Transmission using TCP/IP; **Networking Models & Applications:** Centralized, Decentralized, and Distributed; Client-Server and Peer-to-Peer; File sharing & Web- based; Content Distribution Net-	15

	works;
	Introduction to Example Networks: The Internet and its Conceptual View ; Internet Services; Accessing The Internet; Connection-Oriented Networks: X.25, Frame Relay and ATM;	

II	**Data Communication Concepts & Components**: Digital and Analog Data and Signals, Asynchronous and Synchronous transmission; bit rate & baud, bandwidth & Channel Capacity; Nyquist Bit Rate, Shannon Capacity; Network Performance Parameters; Transmission Impairment;
	Connecting Devices & Transmission Media: Network Interface Cards, Connectors, Hubs, Transceivers & Media Connectors; Link-Layer Switches, Bridge, Routers, Gateways, Virtual LANs; Guided Transmission Media; Wireless transmission; Satellite communication;
	Data Encoding & Modulation Techniques: NRZ, NRZ-I, Manchester and Differential Manchester encoding; 4B/5B ; Pulse Code Modulation & Delta Modulation; Digital to Analog encoding;
	Switching and Bandwidth Utilization: Methods of Switching; Virtual Circuit & Datagram Networks; Multiplexing; Spread Spectrum;
	Wired Networks and the Local Loop: Telephone Networks; Modems; Broadband and ADSL; ADSL Versus Cable; Hybrid Fiber-Coaxial Network ; Fiber-to-the-Home Broadband;
		15

III	**Data Link Layer**: Communication at the Data Link Layer; Nodes and Links; Link Layer Addressing; Examples of Data Link layer protocols;
	Design Issues: Framing techniques; Error Detection and Correction; Sliding Window Flow Control Protocols;
	Media Access Control: Random Access: Aloha, CSMA , CSMA/CD; Collision free protocols with Controlled Access; Wavelength Division Multiple access for Fiber-Optic Data Communication;
	IEEE LAN standards: Ethernet (Physical specifications, Encoding, Frame Format & MAC protocol); Binary Exponential Backoff algorithm;
	Introduction to Wireless Networks: IEEE 802.11 Wireless LAN; Wi-Max; Wireless LAN Protocol: MACA; Bluetooth and other wireless PAN technologies; Cellular Networks: Generations; GSM, CDMA, LTE.
		15

IV	**Transport layer** : Addressing, Services and Protocols; TCP and UDP services & header formats;
	Network Layer : Services, Routing Algorithms: Shortest Path Routing, Flooding , Distance Vector Routing, Link State Routing, Hierarchical Routing, Multi Cast Routing, Routing for Mobile hosts;
	Network Layer in TCP/IP: Basic characteristics of IP protocol; addressing and header format of IPv4 ; IPv6;
	Congestion Control & Quality of Service: General Principals; Congestion control in Virtual – Circuit Subnets; Congestion Control in Datagram Subnets: Choke packets, Load Shedding; Random Early Detection, Jitter Control; Over provisioning, Buffering, Traffic Shaping, Leaky Bucket, Token Bucket, Resource Reservation, Admission Control, Packet Scheduling;
		15

	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Andrew S. Tanenbaum, Computer Networks, 4th Edition - PHI.
2) Behrouz A Forouzan, Data Communications and Networking, 5th Edition- Mc-Graw Hill Education.
3) Michael A. Gallo, William M. Hancock, Computer Communications and Networking Technologies – CENGAGE learning.
4) William Stallings, Data and Computer Communications, 5th Edition – PHI.

CC-7 Database Management Systems

With effect from the Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	2nd
Name of the Course	Database Management Systems
Course Code	M24-CAP-203
Course Type	CC-7
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a comprehensive understanding of database concepts, including the three-schema architecture, relational models, and the ER model. It covers SQL and PL/SQL for database management, exploring queries, constraints, and advanced functions. Students will learn relational algebra, normalization techniques, and query optimization to enhance database design and performance. The course also addresses transaction processing, concurrency control, and database recovery, emphasizing reliability, consistency, and security in database systems.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1**: Understand and apply the three-schema architecture, data independence, and entity-relationship modeling to design effective database schemas. **CLO-2**: Develop and execute SQL and PL/SQL queries, including advanced operations like joins, constraints, triggers, and aggregate functions, for robust database management. **CLO-3**: Analyze relational algebra operations and apply normalization techniques to optimize database structure and ensure data integrity. **CLO-4**: Demonstrate knowledge of transaction processing, concurrency control methods, and database recovery techniques to maintain database reliability and security.

	Theory	Practical	Total
Credits	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Database System Concepts and Architecture: Three - Schema Architecture and Data Independence, Entity Relationship Model: Entity Types, Entity Sets, Attributes & keys, Relationships Types & instances ER Diagrams, Naming conventions and Design Issues. Relational Model Constraints, Concept of Keys.	15
II	SQL: Data Definition and Data Types, DDL, DML, and DCL, Join Operations, Views & Queries in SQL, Specifying Constraints & Indexes in SQL, aggregate functions - min, max, count, average, sum. Group by, Order by and Having clauses, PL/SQL: Architecture of PL/SQL, Basic Elements of PL/SQL, PL/SQL Transactions, Cursors and Triggers.	15

III	Relational Algebra: Unary and Binary Relational Operations, Functional Dependencies, Normal Forms Based on Primary Keys- (1NF, 2NF, 3NF, BCNF), Multi-valued Dependencies, 4 NF, Join dependencies, 5 NF, Domain Key Normal Form. Query Processing and Optimization	15
IV	Transaction Processing Concepts: Introduction to Transaction Processing, Transaction & System Concepts, Properties of Transaction, Schedules and Recoverability, Serializability of Schedules. Concurrency Control Techniques: Locking Techniques, Time stamp ordering, Multi-version Techniques, Database backup, recovery and security.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Date C.J., An Introduction to Database Systems, Pearson Education.
2) Hector G.M., Ullman J.D., Widom J., Database Systems: The Complete Book, Pearson Education.
3) Silberschatz A., Korth H., Sudarshan S., Database System Concepts, McGraw Hill.

CC-8 Artificial Intelligence

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	2nd		
Name of the Course	Artificial Intelligence		
Course Code	M24-CAP-204		
Course Type	CC-8		
Level of the course (As per Annexure-I	400-499		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a comprehensive introduction to the concepts, theories, and applications of Artificial Intelligence (AI), enabling students to understand various knowledge representation techniques using propositional logic, predicate logic, and fuzzy logic. It also introduces search techniques for problem-solving, covering uninformed, informed, and game-playing strategies. Additionally, the course explores the functioning of production systems, expert systems, genetic algorithms, and machine learning techniques, offering students practical insights into their applications in AI.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1 Demonstrate an understanding of the foundational concepts of AI, its historical development, and the distinction between Strong AI and Weak AI. CLO-2 Apply propositional logic, predicate logic, and fuzzy logic for knowledge representation and reasoning in AI systems. CLO-3 Implement various search algorithms such as BFS, DFS, A*, and Minimax to solve complex AI problems, including two-player games. CLO-4 Explain and apply machine learning algorithms including supervised (e.g., neural networks, decision trees) and unsupervised (e.g., k-means, PCA) techniques for data analysis and AI applications.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Definition, history, and evolution of AI, Strong AI vs. Weak AI, Applications of AI; Knowledge Representation using logic: Propositional logic: syntax, semantics, truth tables, logical connectives, inference rules, Predicate logic: first-order logic, quantifiers, predicates, clausal form and unification; Fuzzy logic: fuzzy sets, membership functions, fuzzy reasoning.	15
II	Search Techniques: Problem formulation: state space representation, Uninformed Search Strategies: Breadth-First Search, Depth-First Search (DFS), Iterative Deepening DFS; Informed Search Strategies: Hill climbing, Best-first search, A* algorithm, admissibility,	15

	monotonicity, and informedness, Search in Two-Player Games: Minimax algorithm, Alpha-Beta pruning.	
III	Production Systems: rules, working memory, and control strategies, forward chaining and backward chaining, commutative and non-commutative production systems, Expert Systems: Definition and characteristics, Architecture, Applications; Genetic Algorithms: Components of GAs: chromosomes, crossover, mutation, selection, replacement, Fitness functions and evolution processes, GA vs. traditional problem-solving techniques	15
IV	Machine Learning (ML): Definition and importance, Types: supervised, unsupervised, reinforcement learning; Supervised Learning: Linear regression, Decision Trees, k-Nearest Neighbors (k-NN), Neural networks: introduction, perceptron, multilayer networks, back-propagation, Unsupervised Learning: Algorithms: k-Means clustering, Hierarchical clustering, Principal Component Analysis.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Luger, G. F. (2009). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (6th ed.). Pearson Education.
2) Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.
3) Rich, E., Knight, K., & Nair, S. B. (2017). *Artificial Intelligence* (3rd ed.). McGraw-Hill Education.
4) Coppin, B. (2004). *Artificial Intelligence Illuminated*. Narosa Publishing House.

PC-3 PRACTICAL-4

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	2^{nd}
Name of the Course	Practical-4
Course Code	M24-CAP-205
Course Type	PC-3
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This course aims to provide hands-on experience in building web applications and understanding networking concepts. Part A focuses on mastering server-side development with Node.js and Express.js, enabling students to design efficient applications integrated with databases like MongoDB. Part B emphasizes networking principles, offering insights into data transmission, error detection, routing, and network protocols through programming and simulation, preparing students for real-world applications in web development and network administration.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** and implement server-side applications using Node.js and Express.js, including handling HTTP methods, managing file operations, and building RESTful APIs integrated with MongoDB for CRUD operations and authentication. **CLO-2:** Demonstrate the ability to use Node.js core modules, custom modules, middleware, and debugging tools to build dynamic, efficient, and error-resilient web applications. **CLO-3:** Analyze and implement networking concepts and protocols, including OSI and TCP/IP models, socket programming, and data transmission methods, using Python or C++. **CLO-4:** Apply algorithms and techniques in networking, such as error detection (CRC), routing (Dijkstra's algorithm), and flow control protocols (Go-Back-N, Selective Repeat), through programming and simulation tools.

Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 3 questions from the Part-A and 2 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and one from the Part-B.	120
Part-A 1) Set up a simple HTTP server in Node.js that responds with "Hello, World!" when accessed via a browser. 2) Illustrate the client-server architecture by creating a basic web application that sends a request to the server and displays the response in the browser. 3) Implement a program to demonstrate the request-response cycle by logging HTTP request headers and returning a JSON response. 4) Compare server-side and client-side operations by creating a simple application where the server processes data and the client displays it. 5) Install Node.js and initialize a new project using npm. Create and manage	60

packages using `package.json`. 6) Write a program using Node.js core modules like `fs` and `os` to read system information and save it to a file. 7) Create a custom module for string manipulation (e.g., reversing, converting to uppercase) and use it in a Node.js script. 8) Write a Node.js program to read and write data to a text file asynchronously, logging success or error messages to the console. 9) Create a script to list all files and directories in a specified folder and display them hierarchically. 10) Implement a program that manages a directory: creating it if it doesn't exist, adding files, and deleting files. 11) Build a basic HTTP server in Node.js that supports different HTTP methods (GET, POST, PUT, DELETE) and logs each request. 12) Create a server that serves static files (e.g., HTML, CSS, JS) from a public directory. 13) Use the `EventEmitter` class to create and emit custom events, such as notifying users when a file operation is completed. 14) Implement a real-time data handler using events, simulating a live stock ticker system. 15) Create a script that performs file operations and uses `try-catch` blocks to handle file-not-found errors gracefully. 16) Debug a Node.js script using `node --inspect` and Chrome DevTools, identifying and fixing a logical error. 17) Create a basic Express.js application to handle routing for `/home`, `/about`, and `/contact` with respective responses. 18) Develop a RESTful API using Express.js to manage a list of books (CRUD operations). 19) Set up a server-side rendering engine (EJS) to dynamically generate HTML pages with user data. 20) Implement custom middleware in an Express.js application to log request details and handle errors. 21) Implement a JWT-based authentication system for a RESTful API, allowing users to register and log in. 22) Create an Express.js application to demonstrate session management for user login and logout. 23) Install MongoDB locally and create a database called `school`. Add a `students` collection and insert sample documents. 24) Use MongoDB Atlas to create a cloud-hosted database and connect to it using Node.js. 25) Write a script to query MongoDB for documents with specific conditions, such as retrieving students with grades above 80. 26) Develop a Node.js script to perform CRUD operations on a `products` collection in MongoDB. 27) Create an Express.js application that connects to MongoDB and exposes APIs for CRUD operations on a `tasks` collection. 28) Add indexes to a MongoDB collection and demonstrate their impact on query performance by measuring execution time before and after indexing. 29) Write a script that creates a compound index on multiple fields in a collection and tests its effectiveness with specific queries.	
Part-B 1) Compare the OSI and TCP/IP reference models by creating a document that maps the functionality of each layer. 2) Develop a Python script to simulate data transmission using TCP/IP sockets between a client and server. 3) Write a program to calculate Nyquist Bit Rate and Shannon Capacity for a given set of inputs (bandwidth, signal levels, noise). 4) Implement 4B/5B encoding for a given binary sequence using Python or C++. 5) Implement time-division multiplexing for multiple signals using Python.	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by

6) Compare ADSL and cable broadband connections by analyzing speed, latency, and reliability. 7) Simulate and test the operation of sliding window flow control protocols (Go-Back-N and Selective Repeat). 8) Write a program to implement error detection using CRC (Cyclic Redundancy Check). 9) Implement the Binary Exponential Backoff algorithm and simulate its role in collision resolution. 10) Implement a shortest path routing algorithm (e.g., Dijkstra’s algorithm) to find the optimal path in a simulated network.	students.)

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ Practicum	**30**	**➢ Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) "Node.js Design Patterns" by Mario Casciaro and Luciano Mammino
2) "Learning Node.js Development" by Andrew Mead
3) "Express in Action" by Evan M. Hahn
4) "REST API Development with Node.js" by Fernando Doglio
5) "MongoDB: The Definitive Guide" by Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow
6) "Learning MongoDB" by Amit Phaltankar, Juned Ahsan, and Michael Harrison
7) Andrew S. Tanenbaum, Computer Networks, 4th Edition - PHI.
8) Behrouz A Forouzan, Data Communications and Networking, 5th Edition- Mc-Graw Hill Education.

PC-4 PRACTICAL-5

With effect from Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	2nd		
Name of the Course	Practical-5		
Course Code	M24-CAP-206		
Course Type	PC-4		
Level of the course	400-499		
Pre-requisite for the course (if any)			
Course objectives	The primary objective of this course is to equip students with the theoretical knowledge and practical skills necessary to solve complex computational problems. Through Part A, students will gain expertise in database design, implementation, and optimization, leveraging advanced SQL and PL/SQL techniques. Part B focuses on problem-solving using search algorithms, state-space representations, and optimization methods such as Genetic Algorithms, enabling students to tackle challenges in AI and operations research efficiently.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Analyze, design, and implement database schemas for real-world applications such as library management, hospital management, and e-commerce systems, while applying concepts of ER diagrams, relational schema, normalization, and functional dependencies. **CLO-2:** Demonstrate the ability to write and optimize SQL queries, implement advanced database features such as views, indexes, triggers, cursors, and perform database operations using relational algebra and PL/SQL programming. **CLO-3:** Formulate and implement state-space representations, search algorithms (BFS, DFS, Iterative Deepening DFS), and production systems to solve complex problems like puzzles, mazes, and water-jug problems. **CLO-4:** Design and develop solutions using Genetic Algorithms by encoding chromosomes, defining fitness functions, and applying these techniques to solve optimization problems like the Travelling Salesman Problem (TSP) and mathematical function maximization.		
Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	
Part B- Contents of the Course			
Practicals			**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 3 questions from the Part-A and 2 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to write and execute 2 questions from the Part-A and one from the Part-B.			120
Part-A 1) Create an ER diagram for a library management system that includes entity types, attributes, keys, relationships, and instances. 2) Convert the ER diagram into relational schemas and define the primary and foreign keys. 3) Implement a database schema in a DBMS for an e-commerce application. Define the constraints, such as NOT NULL, UNIQUE, CHECK, and FOREIGN KEY. 4) Create a database for a hospital management system. Define tables for doctors, patients, appointments, and prescriptions.			60

5) Perform basic operations such as inserting, updating, and deleting records. 6) Write queries to retrieve data from multiple tables using INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. 7) Create a query to find patients who have visited a specific doctor using JOIN. 8) Create a view to display the total number of patients attended by each doctor. 9) Add an index to optimize the search for patients by their last names. 10) Write a PL/SQL program to implement a banking transaction system that transfers money between two accounts. Use COMMIT and ROLLBACK statements. 11) Create a cursor to fetch and display all overdue book records from a library database. 12) Develop a trigger to automatically update the stock count when a new product is added to an inventory database. 13) Write and execute queries in relational algebra for the following operations: selection, projection, union, intersection, difference, Cartesian product, and join for a student database. 14) Identify functional dependencies in a given database (e.g., a university database). 15) Normalize the database to 1NF, 2NF, 3NF, and BCNF, showing each step of decomposition. 16) Write an inefficient query for fetching data from a large database. Use EXPLAIN PLAN to analyze it and optimize the query using indexes and appropriate joins.	
Part-B 1) Formulate a state-space representation for the "8-puzzle problem." Represent states, actions, and transitions clearly and define the goal state. 2) Implement Breadth-First Search (BFS) to solve a maze where the start and goal positions are specified. 3) Use Depth-First Search (DFS) to navigate through a graph of cities and find a path from a given source to a destination. 4) Apply Iterative Deepening DFS to solve a water-jug problem (e.g., measure exactly 4 liters using a 3-liter and a 5-liter jug). 5) Develop a production system to solve the "Tower of Hanoi" problem. 6) Write a program to implement a Genetic Algorithm to maximize a mathematical function (e.g., f(x)=x2,0≤x≤31). Demonstrate the use of binary encoding for chromosomes. 7) Define a fitness function for solving the "Travelling Salesman Problem (TSP)" using a Genetic Algorithm.	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Silberschatz A., Korth H., Sudarshan S., Database System Concepts, McGraw Hill.
2) Luger, G. F. (2009). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (6th ed.). Pearson Education.

BC-3 Mathematical Foundations for Computer Science

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA		
Semester	2nd		
Name of the Course	Mathematical Foundations for Computer Science		
Course Code	M24-CAP-207		
Course Type	BC-3		
Level of the course (As per Annexure-I	400-499		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this paper is to make the students familiar with the commonly used mathematics and statistics in the field of computer science.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will be able to apply set theory, relations, and functions to solve problems in discrete mathematics, including the use of pigeonhole principles and recursive functions. **CLO-2:** Students will demonstrate proficiency in performing matrix operations, solving systems of linear equations, and applying numerical methods for interpolation, integration, and differentiation. **CLO-3:** Students will develop the ability to organize, analyze, and interpret data using measures of central tendency, dispersion, and statistical visualization techniques. **CLO-4:** Students will gain the ability to model relationships between variables using regression and correlation analysis, and apply probability principles, including Bayes' theorem, to real-world scenarios.		
Credits	Theory	Practical	Total
	0	0	0
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

<u>Instructions for Paper- Setter:</u> The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Sets: Set theory: Basic concept, set types, set operations, cardinality, and notation. Relations: Relations and its representations, Properties of binary relation –Reflexive, symmetric, Asymmetric, transitive, Equivalence, Inverse & Composition of a relation, closure of relations, its types, Partial ordering relation, Hasse diagram, minimal elements, upper bound, lower bound, Lattices Functions: definition, floor functions, ceiling functions, surjective, injunctive and bijective functions, Inverse Function, Composition of functions, recursive Functions, Pigeon hole principles and its application.	15
II	Addition and multiplication of matrices, Laws of matrix algebra, Singular and non-singular matrices, Inverse of a matrix, Systems of linear equations, Eigen values and Eigen vectors, Diagonalization of a square matrix. Interpolation, Numerical Integration and Differentiation.	15
III	Statistical Methods: Definition and scope of Statistics, concepts of statistical population and sample. Data: Quantitative and qualitative, attributes, variables, scales of measurement nominal, ordinal, interval and ratio. Presentation: tabular and graphical, including histogram	15

	and ogives. Measures of Central Tendency: Mean, Median, Mode. Measures of Dispersion: range, quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, skewness and kurtosis. Statistical Methods: Definition and scope of Statistics, concepts of statistical population and sample. Data: Quantitative and qualitative, attributes, variables, scales of measurement nominal, ordinal, interval and ratio. Presentation: tabular and graphical, including histogram and ogives. Measures of Central Tendency: Mean, Median, Mode. Measures of Dispersion: range, quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, skewness and kurtosis.	
IV	Bivariate data: Definition, scatter diagram, simple, partial and multiple correlation (3 variables only), rank correlation. Simple linear regression, principle of least squares and fitting of polynomials and exponential curves. Probability: Introduction, random experiments, sample space, events and algebra of events. Definitions of Probability – classical, statistical, and axiomatic. Conditional Probability, laws of addition and multiplication, independent events, theorem of total probability, Bayes' theorem and its applications.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Gupta, S. C. and Kapoor, V.K. : Fundamentals Of Mathematical Statistics, Sultan Chand &Sons
2) Seymour Lipschutz, Marc Lars Lipson, Discrete mathematics, McGraw-Hill international editions, Schaum's series.
3) V. Rajaraman, Computer-Oriented Numerical Methods., PHI Reference Books:
4) Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
5) Hogg, R.V., Tanis, E.A. and Rao J.M. : Probability and Statistical Inference, Seventh Ed, Pearson Education, New Delhi.
6) Goon A.M., Gupta M.K. and Dasgupta B. (2002): Fundamentals of Statistics, Vol. I & II, The World Press, Kolkata.
7) Babu Ram: Discrete Mathematics
8) Shanti Narayana : Differential & Integral calculus

BC-4 PRACTICAL-6

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-6
Course Code	M24-CAP-208
Course Type	BC-4
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	The course aims to equip students with the ability to implement fundamental concepts of sets, relations, and functions using programming techniques. It focuses on developing skills for performing matrix operations, solving linear equations, and applying numerical methods through programming. Additionally, it enables students to analyze statistical data, apply probability concepts, and perform regression analysis using computational tools.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will be able to implement fundamental concepts of set theory, relations, and functions, including operations, properties, and representations, to solve real-world problems using programming techniques. **CLO-2:** Students will be able to perform matrix operations, solve systems of linear equations, and apply numerical techniques such as interpolation, integration, and differentiation through programming exercises. **CLO-3:** Students will be able to analyze and represent data using statistical measures, including measures of central tendency, dispersion, and graphical tools, and interpret the results effectively. **CLO-4:** Students will be able to apply probability theories, Bayes' theorem, and regression analysis to model and solve problems involving uncertainty and bivariate data using programming tools.

Credits	Theory	Practical	Total
	0	0	0
Teaching Hours per week	0	4	4
Internal Assessment Marks	0	15	15
End Term Exam Marks	0	35	35
Max. Marks	0	50	50
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	**Contact Hours**
The examiner will set 3 questions at the time of practical examination by taking course learning outcomes (CLO) into consideration. The examinee will be required to write and execute 2 questions.	60
1) Set Operations: Write a program to perform union, intersection, difference, and symmetric difference on two sets. 2) Cardinality: Implement a program to calculate the cardinality of a given set. 3) Binary Relation Properties: Write a program to check whether a given binary relation is reflexive, symmetric, asymmetric, or transitive. 4) Hasse Diagram: Create a program to generate a Hasse diagram for a partial ordering relation. 5) Lattices: Write a program to verify whether a given set with a partial order forms a lattice. 6) Functions: Implement a program to check whether a function is injective,	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

surjective, or bijective.

7) Recursive Functions: Write a program to compute the factorial of a number using recursion and apply it to solve problems using pigeonhole principles.
8) Pigeonhole Principle: Write a program to prove the pigeonhole principle for a given input set.
9) Matrix Operations: Write a program to add, subtract, and multiply two matrices.
10) Inverse of a Matrix: Implement a program to find the inverse of a square matrix using Gauss-Jordan elimination.
11) Eigenvalues and Eigenvectors: Write a program to compute the eigenvalues and eigenvectors of a square matrix.
12) System of Linear Equations: Create a program to solve a system of linear equations using Gaussian elimination.
13) Diagonalization: Write a program to diagonalize a square matrix if possible.
14) Interpolation: Implement a program to perform Lagrange or Newton interpolation for a given set of points.
15) Numerical Integration: Write a program to compute the definite integral of a function using the trapezoidal or Simpson's rule.
16) Numerical Differentiation: Create a program to find the derivative of a function using finite difference methods.
17) Data Presentation: Write a program to create a histogram and ogive for a given data set.
18) Measures of Central Tendency: Implement a program to calculate mean, median, and mode for a given data set.
19) Measures of Dispersion: Write a program to compute range, quartile deviation, mean deviation, standard deviation, and coefficient of variation for a data set.
20) Moments, Skewness, and Kurtosis: Create a program to calculate the moments of a distribution and determine its skewness and kurtosis.
21) Tabular Representation: Write a program to present data in tabular form based on user input (quantitative or qualitative).
22) Scatter Diagram: Write a program to generate a scatter plot for bivariate data and compute the correlation coefficient.
23) Regression: Implement a program to compute the equation of a simple linear regression line and predict values based on the model.
24) Polynomial Fitting: Write a program to fit a polynomial curve using the principle of least squares.
25) Exponential Curve Fitting: Create a program to fit an exponential curve to a given data set.
26) Probability Calculations: Write a program to compute probabilities using classical, statistical, and axiomatic definitions.
27) Conditional Probability: Implement a program to calculate conditional probability and verify the laws of addition and multiplication.
28) Bayes’ Theorem: Write a program to solve problems using Bayes' theorem.
29) Random Events Simulation: Create a program to simulate random experiments, generate a sample space, and calculate probabilities.

Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Practicum**	**15**	➢ **Practicum**	**35**
• Class Participation:	4	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	4		
• Mid-Term Examination:	7		

Part C-Learning Resources

Reference Books:

1) Gupta, S. C. and Kapoor, V.K. : Fundamentals Of Mathematical Statistics, Sultan Chand &Sons
2) Seymour Lipschutz, Marc Lars Lipson, Discrete mathematics, McGraw-Hill international editions, Schaum's series.
3) V. Rajaraman, Computer-Oriented Numerical Methods., PHI Reference Books:
4) Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
5) Hogg, R.V., Tanis, E.A. and Rao J.M. : Probability and Statistical Inference, Seventh Ed, Pearson Education, New Delhi.
6) Goon A.M., Gupta M.K. and Dasgupta B. (2002): Fundamentals of Statistics, Vol. I & II, The World Press, Kolkata.
7) Babu Ram: Discrete Mathematics
8) Shanti Narayana : Differential & Integral calculus