Master of Computer Applications
Semester–I

Course Code	Course Title	L	T	P	Credits
MCA-101A	**Introduction to Web Technology** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-101A.1	Design static web pages using HTML and CSS with multimedia and advanced styling.
MCA-101A.2	Utilize JavaScript for DOM manipulation, input validation, event handling, and enhancing interactivity.
MCA-101A.3	Apply PHP server-side: database, forms, sessions, dynamic content, and connectivity.
MCA-101A.4	Apply best practices for secure AJAX and accessible web applications.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-101A.1	3	2	3	3	-	-	–	1
MCA-101A.2	3	2	3	3	-	–	–	1
MCA-101A.3	3	2	3	3	-	-	–	1
MCA-101A.4	2	2	3	3	-	-	2	1

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. 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 – I — Contact Hours: 11

Fundamental concepts of HTML and CSS

Fundamentals of web browsers and their functions, web optimizations, static web page design using HTML 5.0 including multimedia, graphics, and form tags. Cascading Style Sheet (CSS) concepts and properties such as borders, backgrounds, fonts, and text effects. Buffering, weblogs, and web cache poisoning.

UNIT – II — Contact Hours: 12

JavaScript programing concepts

JavaScript, focusing on the Document Object Model (DOM), user inputs, memory concepts, operators, control structures, and looping constructs.

Programmer-defined functions, scoping rules, recursion, and iteration, array declaration and allocation, and passing arrays to functions.

JavaScript objects such as String, Date, Boolean, Window, and document, cookies and perform form validation in JavaScript. Event handling using JavaScript.

UNIT – III **Contact Hours: 10**

Installation of MySQL and dynamic web page designing with PHP

Install and configure MySQL and PHP, basic security guidelines, variables, data types, operators and expressions, constants, and flow control functions. Switching flow, loops, code blocks, and browser output, cookies, sessions, and dynamic content generation using PHP.

UNIT – IV **Contact Hours: 12**

Designing AJAX applications

AJAX and its role in modern web technologies. Creating simple AJAX applications and interacting with web servers using the XML HttpRequest object. AJAX and non-AJAX applications, working with PHP and AJAX to process client requests and access files using PHP, security and accessibility in AJAX applications, including secure AJAX applications and accessible rich internet applications.

Text Books:

1. Deitel H.M., Deitel P.J., *Internet & World Wide Web: How to program*, Pearson Education.
2. Kogent Learning, *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book*, Wiley India Pvt. Ltd.

Reference Books:

1. Boronczyk, Naramore, *Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Thomas Powell, Ajax: *The Complete Reference Book*.

Course Code	Course Title	L	T	P	Credits
MCA-171A	**Web Technology Lab** (Pre-requisite: HTML, CSS, Java Script, PHP,MySQL,Ajax)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-171A.1	Develop and Style Web Pages Using HTML5 and CSS
MCA-171A.2	Utilize JavaScript and Web Security Measures
MCA-171A.3	Develop Server-Side Applications Using PHP and MySQL
MCA-171A.4	Create Accessible and Interactive Web Applications using AJAX

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-171A.1	3	-	3	3	-	-	–	2
MCA-171A.2	3	2	-	3	-	–	2	2
MCA-171A.3	3	3	3	3	-	2	–	2
MCA-171A.4	3	3	3	3	-	2	2	2

List of Experiments

No.	Experiment Detail
1.	Create a basic HTML5 webpage that includes text, images, and links.
2.	Design a multimedia-rich HTML5 webpage incorporating audio, video, and graphics.
3.	Develop a static web page using HTML5 form tags to create a registration form
4.	Apply CSS 2.0 properties to style an HTML5 webpage with different selectors, classes, and IDs.
5.	Utilize CSS 3.0 properties to enhance an HTML5 webpage with borders, backgrounds, fonts, and text effects.
6.	Implement a simple buffering mechanism and analyze the performance impact on web pages.
7.	Create and analyze weblogs to understand web traffic and user behavior.
8.	Demonstrate web cache poisoning and implement basic security measures to prevent it.
9.	Create a JavaScript program to manipulate the DOM and update webpage content dynamically.
10.	Develop a JavaScript-based form that validates user inputs and provides real-time feedback.
11.	Write JavaScript functions to perform various operations (recursion, iteration, array manipulation).
12.	Implement JavaScript event handling to manage user interactions such as mouse clicks and keyboard events.
13.	Create a JavaScript program to manage cookies and demonstrate their use in maintaining user sessions.
14.	Install and configure MySQL and PHP on a local server, following security best practices.

15.	Develop a PHP script to connect to a MySQL database, perform CRUD operations, and display results on a webpage.
16.	Create a dynamic PHP webpage that uses sessions and cookies to track user activity.
17.	Build a simple AJAX application that retrieves data from a server and updates the webpage without reloading.
18.	Develop an AJAX-based form that submits data to a server-side PHP script and displays the response dynamically.
19.	Implement a secure AJAX application that handles user authentication and data access.
20.	Create an accessible rich internet application using AJAX and PHP, ensuring compatibility with assistive technologies.

Text Books:

1. Deitel H.M., Deitel P.J., *Internet & World Wide Web: How to program*, Pearson Education.
2. Kogent Learning, *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book*, Wiley India Pvt. Ltd.

Reference Books:

1. Boronczyk, Naramore, *Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Thomas Powell, Ajax: *The Complete Reference Book*.

Course Code	Course Title	L	T	P	Credits
MCA-103A	**Advanced Data Structures using C++** (Pre-requisite: C++)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
MCA-103A.1	Learn about analysing algorithms and understand and apply the array data structure along with various operations on it, sorting and searching
MCA-103A.2	Understand and apply the concepts of linked list, stacks and queue data structures
MCA-103A.3	Understand and apply the tree data structure in various fields
MCA-103A.4	Design and analyze the algorithms for graph and hashing

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-103A.1	3	3	–	-	–	–	–	2
MCA-103A.2	3	–	-	-	–	–	–	2
MCA-103A.3	3	2	2	2	–	–	–	2
MCA-103A.4	3	3	3	2	–	–	–	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. 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 – I **Contact Hours: 12**

Data structures, algorithm complexity, searching, sorting techniques and implementation

Classification of Data Structures, Analyzing Algorithms, Time and Space Analysis of Algorithms, Abstract Data Types, Arrays, Representation of Arrays in Memory, Operations on Array and its Representation in Memory, Pointers, Sparse Matrices. Sorting: Bubble Sort, Selection Sort, Insertion Sort, Merge sort, Radix sort, Bucket sort. Searching: Linear Search (Iterative and Recursive), Binary Search (Iterative and Recursive). Implementation of Arrays, String, Sorting and Searching in C++

UNIT – II **Contact Hours: 11**

Linked lists, Stacks, Queues data structures, operations, applications and implementation

Linked Lists: Introduction, Types (singly, double, circular, circular double) and Operations (Insertion, Deletion, Traversal, Searching, Sorting), Applications, Dynamic Memory Management, Polynomial Representation and Addition, Implementation of Linked Representations in C++. Stacks & Queues: Representation of Stacks, Stack Operations, Applications, Recursion, Queues, Operations on Queues, Circular Queues, Dequeue, Priority Queues, Applications, Implementation of Stacks and Queues in C++

UNIT – III **Contact Hours: 12**

Various forms of Trees data structures, operations, applications and implementation

Definition and Basic Terminologies, Representation of Trees, Binary tree,, Representation of Binary trees , Pre-order, In-order, Post-ordered traversal. Recursive Algorithms. Traversal methods. Representation of trees and its applications: Binary tree representation of a general tree. Conversion of forest into tree. Threaded binary trees. Binary search tree: Height balanced (AVL) tree, B-trees, M-Way Search Trees, B+ Trees, Splay tree. Applications and Implementation of trees in C++.

UNIT – IV **Contact Hours: 10**

Various forms of Graph data structures, operations, applications and implementation

Representation, Type of Graphs, Paths and Circuits: Euler Graphs, Depth-and breadth-first traversals, Minimum Spanning Tree: Prim's and Kruskal's algorithms, Shortest-path Algorithms: Dijkstra's and Floyd's algorithm, Topological sort, Maxflow: Ford-Fulkerson algorithm, max flow –min cut.

Hashing: Hash functions, Collision Resolution, Implementation using Linear and Quadratic Probing, Chaining using C++.

Text Books:

1. G.A.V Pai, *Data Structures and Algorithms*, Tata McGraw-Hill.
2. Drozdek, *Data Structure and Algorithms in C++*, Cengage Learning.

Reference Books:

1. Seymour Lipschutz, *Data Structures*, Tata McGraw-Hill, Schaum's Outlines, New Delhi.
2. Weiss, *Data Structures and Algorithm Analysis in C++*, Pearson Education.
3. Goodrich, *Data Structures and Algorithms in C++*, Wiley India Pvt. Ltd.
4. S. Sahni, Data structures, Algorithms, and Applications in C++, University Press (India) Pvt. Ltd.
5. Walter Savitch, Problem solving with C++, Pearson education.
6. John R. Hubbard, Data Structures with C++, Tata McGraw-Hill, Schaum's Outlines, New Delhi.

Course Code	Course Title	L	T	P	Credits
MCA-173A	**Advanced Data Structures Lab** (Pre-requisite: C++)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-173A.1	Implement and Manipulate Fundamental Data Structures
MCA-173A.2	Apply Stack and Queue Operations for Efficient Data Management
MCA-173A.3	Construct and Operate on Binary Trees and Heaps
MCA-173A.4	Solve Complex Graph Problems Using Advanced Algorithms

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-173A.1	3	2	-	2	-	-	-	2
MCA-173A.2	3	2	2	2	-	-	-	2
MCA-173A.3	3	2	2	2	-	-	-	2
MCA-173A.4	3	3	3	2	-	-	-	2

List of Experiments

No.	Experiment Detail
1.	Write a program that uses functions to perform the following operations on an array i) Creation ii) Insertion iii) Deletion iv) Traversal.
2.	Write a program that uses functions to perform the following operations on strings i) Creation ii) Insertion iii) Deletion iv) Traversal.
3.	Write a menu driven program that implements singly linked list for the following operations: Create, Display, Concate, Merge, Union, Intersection
4.	Write a menu driven program that implements doubly linked list for the following operations: Create, Display, Count, Insert, Delete, Search, Copy, Reverse, Sort
5.	Write a program to implement simple Stack, Queue, Circular Queue, Priority Queue
6.	Write a menu driven program to a. Create a binary search tree b. Traverse the tree in Inorder, Preorder and Post Order c. Search the tree for a given node and delete the node
7.	Write a program to implement insertion and deletion in AVL tree.
8.	Write a program that implements Heap tree (Maximum and Minimum Heap tree) for the following operations. (Using array) Insert, Delete.
9.	Write a program to implement Dijkstra's shortest path algorithm for a directed graph.
10.	Write a program to insert and delete nodes in graph using adjacency matrix.

11.	Write a program to implement Breadth First search using linked representation of graph.
12.	Write a program to implement Depth first search using linked representation of graph.
13.	Write a program to create a minimum spanning tree using Kruskal's algorithm.
14.	Write a program to create a minimum spanning tree using Prim's algorithm.
15.	Write a program to implement Topological sort.

Text Books:

1. G.A.V Pai, *Data Structures and Algorithms*, Tata McGraw-Hill.
2. Drozdek, *Data Structure and Algorithms in C++*, Cengage Learning.

Reference Books:

1. Seymour Lipschutz, *Data Structures*, Tata McGraw-Hill, Schaum's Outlines, New Delhi.
2. Weiss, *Data Structures and Algorithm Analysis in C++*, Pearson Education.
3. Goodrich, *Data Structures and Algorithms in C++*, Wiley India Pvt. Ltd.
4. S. Sahni, Data structures, Algorithms, and Applications in C++, University Press (India) Pvt. Ltd.
5. Walter Savitch, Problem solving with C++, Pearson education.
6. John R. Hubbard, Data Structures with C++, Tata McGraw-Hill, Schaum's Outlines, New Delhi.

Course Code	Course Title	L	T	P	Credits
MCA-105A	**Programming with Java** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-105A.1	Discuss the basic features of Java language.
MCA-105A.2	Construct programs utilizing various object-oriented programming (OOP) principles and packages.
MCA-105A.3	Utilize advanced Java concepts of exceptions handling, files, and multithreading.
MCA-105A.4	Develop GUI applications using AWT Event handling, Swings and database connectivity JDBC.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-105A.1	3	2	2	2	-	-	-	1
MCA-105A.2	3	3	3	2	2	-	-	1
MCA-105A.3	3	2	3	3	1	-	1	2
MCA-105A.4	2	2	3	3	2	1	1	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. 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 – I **Contact Hours: 12**

Fundamentals of Java language

Introduction to Java: Java Features, Java Virtual Machine and bytecode, Basics of Java programming: syntax, variables data types, operators and expressions, statements

Control Flow: Decision-making statements (if, else-if, switch). looping statements (for, while, do-while) and branching statements (break, continue, return),

Arrays: Declaring, initializing and manipulating arrays, array operations.

UNIT – II **Contact Hours: 11**

Object Oriented programing and package creation in Java

Classes and Objects: Declaring Classes and creating Objects, constructors, Garbage Collection, finalize() Method, Access modifiers (public, private, protected, default), static and final modifier, getter and setter methods, ‘this’ keyword, Method overloading, Wrapper Classes.

Inheritance: Extending classes, Method Overriding, ‘super’ keyword, Abstract classes, Multiple Inheritance, Interfaces and Extending Interfaces.

Packages: Java API Packages, importing packages, creating a new packages and using classes from package

UNIT – III **Contact Hours: 12**

Handling Exceptions, working with input/output and multi-threading concepts.

Exception Handling: Types of Errors, Understanding Exceptions, Built-in Exceptions, checked and unchecked exceptions, try-catch block, multiple catch clauses, nested try block, finally block, throw and throws keywords, user created exceptions.

Java I/O Streams**:** Character and Byte streams, Reading console Input using java.util.Scanner and Writing console Output, Reading from and writing to files using FileInputStream, and FileOutputStream, FileReader and FileWriter. Object serialization and de-serialization.

Multithreading: Process versus Threads, Creating threads using Thread class and Runnable interface, thread lifecycle methods, Thread Priorities, synchronization, thread communication.

UNIT – IV **Contact Hours: 10**

Designing GUI applications with event handling and backend database connectivity

GUI and Event Handling: Basics of AWT Classes and Components, Delegation Event Model, Listener interfaces and Adapter classes for Event Handling, Layout Managers (Flow, Border; Grid, Card), Introduction to Java Swing, Components, Containers, JLable, JButton, JFrame, working with Swing Menus.

Java Database Connectivity(JDBC): Types of JDBC Drivers, Key JDBC Classes and Interfaces, Registering JDBC driver, connecting to database, executing SQL queries, processing the ResultSet, closing connection, creating simple JDBC Application.

Text Books:

1. Patrick Naughton, Herbert, Schild, *The Complete reference Java 2*, Tata Mc-Graw Hill.
2. E. Balaguruswamy *Programming with JAVA- A Primer*, Tata Mc-Graw Hill publication

Reference Books:

1. Maydene Fisher, Jon Ellis, Jonathan Bruce, *JDBC API Tutorial and Reference*, Addison Wesley.
2. Giulio Zambon, *Beginning JSP, JSF and Tomcat*, APress

Course Code	Course Title	L	T	P	Credits
MCA-175A	**Programming with Java Lab** (Pre-requisite: Java)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-175A.1	Apply basic java features, control flow, and arrays
MCA-175A.2	Construct object-oriented programs including inheritance, interfaces, and packaging in Java.
MCA-175A.3	Implement exceptions handling, file handling, and multithreading.
MCA-175A.4	Design GUI applications and handle events using AWT, Swings and establish connection with database using JDBC.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-175A.1	3	2	1	2	-	–	–	2
MCA-175A.2	3	2	2	2	–	–	–	2
MCA-175A.3	3	2	3	3	2	-	1	2
MCA-175A.4	3	3	3	3	3	2	2	2

List of Experiments

No.	Experiment Detail
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 operations using arrays and methods.
3.	Write Java program to find the largest and smallest elements in an array.
4.	Implement a Java Program to sort an array of integers using Bubble sort.
5.	Write a program to implement method overloading.
6.	Develop a Java program to implement inheritance by creating a base class Animal and derived classes like Dog and Cat.
7.	Write a Java program to demonstrate method overriding by implementing a base class Shape, and derived classes Circle and Rectangle.
8.	Write a program to handle exceptions using try-catch, multiple catch clauses, and finally block.
9.	Write a program to create and run multiple threads.
10.	Write a program to implement thread life-cycle methods.
11.	Write a program to perform read and write operations on files using FileInputStream and FileOutputStream.
12.	Create a GUI application with AWT components and handle events using delegation event model.

13.	Demonstrate the use of different layout managers and create menus in a GUI application
14.	Create a GUI application with Java Swings
15.	Write a program to connect to a database and perform basic CRUD operations using JDBC.

Text Books:

1. Patrick Naughton, Herbert, Schild, *The Complete reference Java 2*, Tata Mc-Graw Hill.
2. E. Balaguruswamy *Programming with JAVA- A Primer*, Tata Mc-Graw Hill publication

Reference Books:

1. Maydene Fisher, Jon Ellis, Jonathan Bruce, *JDBC API Tutorial and Reference*, Addison Wesley.
2. Giulio Zambon, *Beginning JSP, JSF and Tomcat*, APress

Course Code	Course Title	L	T	P	Credits
MCA-107A	**Operating System and Linux** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
MCA-107A.1	Understand the fundamental concepts, functions, and structures of operating systems, and apply various CPU scheduling algorithms.
MCA-107A.2	Grasp memory concept, allocation technique, paging, segmentation, virtual memory concepts, and file system management.
MCA-107A.3	Learn the history, features and architecture of Linux, perform basic file operations, and write simple shell scripts.
MCA-107A.4	Manage processes, users, and groups in Linux, utilize network commands, perform system administration tasks, and grasp basic security measures.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-107A.1	3	2	–	2	–	–	–	–
MCA-107A.2	3	2	–	–	–	–	–	–
MCA-107A.3	2	-	–	3	–	–	–	-
MCA-107A.4	-	-	-	3	–	–	2	-

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. 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 – I Contact Hours: 11

Fundamentals of Operating System, Process management, CPU scheduling and Deadlock

Introduction to Operating Systems: Definition, types and functions of an Operating system; System structures: Operating system services,, system calls, system programs, and system structures; 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); Deadlocks: Necessary Conditions for Deadlock, Deadlock Prevention, Avoidance, Detection and Recovery.

UNIT – II Contact Hours: 12

Memory management, Paging, Segmentation and Disk scheduling algorithms

Memory Management: Memory Concepts: Logical & Physical Address Space, Swapping, Internal and External Fragmentation; Memory Allocation Techniques: Contiguous Memory Allocation- Single & Multi Partitioned, Non-Contiguous Memory Allocation, Paging and Segmentation. Virtual Memory: Demand paging, page replacement algorithm, allocation of frames, thrashing; File Systems: File concepts; access methods, directory and disk structure, file system mounting, file sharing, protection; Disk Scheduling: algorithms (FCFS, SSTF, SCAN, C-SCAN, LOOK and C-LOOK), Selection of Disk Scheduling Algorithms

UNIT – III **Contact Hours: 10**

Fundamentals of Linux and basic shell script commands

Introduction to Linux: History, features, architecture of Linux; Linus File System: File and directory structure, file permissions, standard file types; basic commands: file and 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.

UNIT – IV **Contact Hours: 12**

Process management, networking and security commands

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, netstar, ssh), configuring network interfaces; System administration: Package management, (installing an removing software using rpm, dpkg, apt-get), backup and restore, logging; Security: basic security concepts, user authentication.

Text Books:

1. Deitel, H.M., *An Introduction to Operating Systems*. Addison Wesley Publishing Company.
2. Milenkovic, M., *Operating Systems - concepts and Design*, McGraw Hill
3. Galvin P., J.L., Abraham Silberschatz. *Operating System Concepts*, John Wiley & Sons Company
4. Tanenbaum, A.S., *Modern Operating System*, Prentice Hall of India Pvt. Ltd.
5. Mokhtar Ebrahim and Andrew Mallett, *Mastering Linux Shell Scripting*, Packt Publishing
6. Love, R., *Linux System Programming*, O'Reilly Media
7. Sobell, M. G., *A Practical Guide to Linux Commands, Editors, and Shell Programming*, Pearson

Reference Books:

1. William Stallings, *Operating Systems*, Prentice Hall of India Pvt. Ltd.
2. Joshi R.C., *Operating System*, Wiley India.
3. Achyut S. Godbole *Operating Systems*, Tata McGraw Hill.
4. D.M. Dhamdhere, *Operating Systems*, Tata McGraw Hill.
5. Das S., *Your UNIX/Linux: The Ultimate Guide*, McGraw-Hill,
6. Nemeth, E. Snyderr, G., Hein, T. R. & Whaley, B., *UNIX and Linux System Administration Handbook*, Pearson
7. Mark Sobell and Matthew Helmke, *A Practical Guide to Linux Commands, and Shell Programming*, Addison-Wesley.

Course Code	Course Title	L	T	P	Credits
MCA-177A	**Operating System and Linux Lab** (Pre-requisite: Linux OS)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

MCA-177A.1	Analyse the performance of different CPU scheduling algorithms and understand the fundamental functions and structures of operating systems.
MCA-177A.2	Design and simulate memory management techniques such as paging and segmentation, and implement basic file system operations like file creation, deletion, reading, and writing.
MCA-177A.3	Develop shell scripts to automate basic system tasks and file operations in Linux, including math operations, system information retrieval, and text processing.
MCA-177A.4	Manage processes, users, groups, and system configurations in Linux, perform file permission handling, basic security checks, and use network and system administration commands.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
MCA-177A.1	3	3	2	2	–	–	–	2
MCA-177A.2	3	2	2	2	–	–	–	2
MCA-177A.3	3	2	2	3	–	–	–	3
MCA-177A.4	3	-	-	3	2	–	3	3

List of Experiments

No.	Experiment Detail
1.	Analyse the performance of different scheduling algorithms.
2.	Design and simulate a memory management system (e.g. paging, segmentation)
3.	Implement a basic file system with operations like file creation, deletion, reading, and writing
4.	Write a script to perform following basic math operations as (i) Addition (ii) Subtraction (iii) Multiplication (iv) Division Note: Take input from i) keyboard ii) command line arguments
5.	Write shell script to show various system configurations like (i) Currently logged user and his long name (ii) Current shell (iii) Your home directory (iv) Your operating system type (v) Your current path setting (vi) Your current working directory (vii) Show all available shells
6.	Write a script to convert the contents of a given file from uppercase to lowercase and also count the number of lines, words and characters of the resultant file. Also display the result in descending order.
7.	Write a shell script to list the files arranged in descending order of their size.
8.	Write a shell script to print the contents of a file from a given line number to next given line number.
9.	Write a shell script to determine whether given file exists or not, file name is supplied as command line

	arguments, also check for sufficient number of command line arguments has been passed.
10.	Write a shell script to delete all lines containing a specific word in one or more file supplied as arguments to it.
11.	A shell script that accepts a list of filenames as its arguments counts and reports the occurrence of each word that is present in the first argument file on other argument files.
12.	Write a shell script to check (i) File Existence (ii) File Permissions (iii) Check if file has execute permission, if not then change permission of file and make it executable (iv) Check whether the file is sorted or not, if not then sort the file (v) Read another file and compare the permissions of both the files, and extract Common permissions if any.
13.	Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th; a\ is to be appended as the indication of folding and the processing is to be continued with the residue. The Input is to be through a text file created by the user.
14.	Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits.
15.	Write a shell script that accepts command line arguments and print them in reverse order.
16.	Write a shell script to check whether the given string is palindrome (i) without using string function (ii) with using string function
17.	Write a shell script to calculate basic salary of Employee on the basis of following conditions: (a) if basic <10000, then hra=15% of basic and da=10% of basic (b) otherwise, hra=25% of basic and da=15% of basic
18.	Write an interactive file handling shell program. Let it offer the user the choice of copying, removing, renaming or linking files. Once the use has made a choice,have the program ask the user for necessary information, such as the file name, new name and so on

Text Books:

1. Deitel, H.M., *An Introduction to Operating Systems*. Addison Wesley Publishing Company.
2. Milenkovic, M., *Operating Systems - concepts and Design*, McGraw Hill
3. Galvin P., J.L., Abraham Silberschatz. *Operating System Concepts*, John Wiley & Sons Company
4. Tanenbaum, A.S., *Modern Operating System*, Prentice Hall of India Pvt. Ltd.
5. Mokhtar Ebrahim and Andrew Mallett, *Mastering Linux Shell Scripting*, Packt Publishing
6. Love, R., *Linux System Programming*, O'Reilly Media
7. Sobell, M. G., *A Practical Guide to Linux Commands, Editors, and Shell Programming*, Pearson

Reference Books:

1. William Stallings, *Operating Systems*, Prentice Hall of India Pvt. Ltd.
2. Joshi R.C., *Operating System*, Wiley India.
3. Achyut S. Godbole *Operating Systems*, Tata McGraw Hill.
4. D.M. Dhamdhere, *Operating Systems*, Tata McGraw Hill.
5. Das S., *Your UNIX/Linux: The Ultimate Guide*, McGraw-Hill,

6. Nemeth, E. Snyderr, G., Hein, T. R. & Whaley, B., *UNIX and Linux System Administration Handbook*, Pearson
7. Mark Sobell and Matthew Helmke, *A Practical Guide to Linux Commands, and Shell Programming*, Addison-Wesley.

Course Code	Course Title	L	T	P	Credits
MCA-179A	**Seminar** (Pre-requisite: None)	0	0	2	1
		CIE	**SEE**		**Total**
		100	0		100

General Guidelines for Seminar:

Seminar is a Skill Enhancement Course (SEC) course that makes students to learn a specific topic through in-depth exploration and analysis of facts about the topic in a set-up that involves presentation, interactive discussions and collaborative learning under the supervision of a teacher. It aims at imparting skills of self-learning, comprehension, communication and presentation, gain knowledge of new technology, exchange ideas, and solve problems.

Topic Finalization:

1. Seminar Topic shall be selected from field of computer science like emerging technologies, latest trends or any research area.
2. Each student shall get the topic approved from a committee of academic in-charge of department and the concerned class in-charge.
3. No two students can choose the same topic for the seminar.
4. Seminar shall be given individually by student.

Execution:

1. After approval, the student will prepare seminar report containing the following and submit the same to the assigned mentor for seminar within one week after declaration of the seminar presentation date by the academic in-charge.
   a. Title
   b. Introduction to the Topic
   c. Literature Review
   d. Technology Used
   e. Suitable Charts, Graphs, Diagrams
   f. Conclusion and Future Scope
   g. Bibliography
2. Each student individually shall present the seminar with the help of power point presentation containing key points, visual aids, animations, videos if required.

Evaluation Criteria:

The seminar has only CIE component of 100 marks that shall be conducted as per Table 1.

Table 1: Relative Weightages of Seminar

Component	Description of the Component		Relative Weightage (Out of 100)
CIE	i)	Seminar Report	20
	ii)	Presentation	20
	iii)	Technical Content	20
	iv)	Depth of Knowledge	20
	v)	Answers to queries	20
		CIE Total	**100**
SEE	NIL		-
		Grand Total	**100**