Master of Computer Application Scheme (2024-2026)

MCA SEM-I Syllabus

Sr No	Category	Course Code	Subject	L	T	P	CA	ESE	Total	Credit
1	PCC	MC1T001	Data Structures & Algorithms	4	0	0	40	60	100	4
2	PCC	MC1T002	Object Oriented Programming using JAVA	3	0	0	40	60	100	3
3	PCC	MC1T003	Operating System	3	0	0	40	60	100	3
4	PCC	MC1T004	Discrete Mathematics & Graph Theory	3	0	0	40	60	100	3
5	AEC	MC1A005	Computer Architecture & Organization	3	0	0	40	60	100	3
6	PCC	MC1L006	Data Structures & Algorithms LAB	0	0	4	60	40	100	2
7	PCC	MC1L007	Object Oriented Programming using JAVA LAB	0	0	4	60	40	100	2
8	PCC	MC1L008	Web Development LAB	0	0	4	60	40	100	2
			Total	**16**	**0**	**12**	**380**	**420**	**800**	**22**

Semester	Course Code	Name of the course	L	T	P	Credits
I	MC1T001	Data Structure & Algorithm	4	0	0	4

Marks Distribution			
CA	**MSE**	**ESE**	**Total**
40	-	60	100

Prerequisites for the course	
1	Basic understanding of programming concepts.

Prior Reading Material/useful links	
1	Data Structures and Algorithm Analysis by Mark Allen Weiss.
2	https://www.geeksforgeeks.org/data-structures/

Course Objectives	
Sr No	**Statement**
1	Introduce the fundamental concepts of data structures and algorithms.
2	Understand and analyze the time and space complexity of algorithms.
3	Learn and implement various data structures such as arrays, stacks, queues, linked lists, trees, and graphs.
4	Apply algorithms for sorting, searching, and hashing.
5	Develop problem-solving skills through the application of data structures and algorithms.

Course Outcomes		
Sr No	**Code**	**CO statement**
1	CO1	Understand and Analyze the Efficiency of Algorithms
2	CO2	Implement and Evaluate Hashing Techniques
3	CO3	Develop and Compare Searching and Sorting Algorithms
4	CO4	Develop and Compare Searching and Sorting Algorithms
5	CO5	Analyze and Implement Tree and Graph Algorithms

Unit No	Course Contents	Duration
I	**Complexity Analysis and Hashing** **Complexity Analysis:** Time and space complexity, asymptotic notations (Big O, Theta, Omega). **Efficient Algorithms:** Importance and performance measurement. **Hashing:** Implementation of dictionaries, hash functions, handling collisions, open addressing, and analysis of search operations.	6 hours

II	**Arrays, Searching, and Sorting** **Arrays:** Abstract Data Type (ADT). **Searching:** Linear and binary search on sorted arrays. **Sorting:** Bubble sort, insertion sort, merge sort, radix sort, bucket sort, and comparison-based sorting models.	7 hours
III	**Stacks and Queues** **Stacks and Queues:** Abstract data types, sequential and linked implementations. **Applications:** Parenthesis matching, Towers of Hanoi, path finding in a maze, queue simulations, and equivalence problems.	7 hours
IV	**Linked Lists** **Linked Lists:** ADT, sequential and linked representations. **Operations:** Insertion, deletion, and search in sequential and linked lists. **Advanced Types:** Doubly linked lists, circular lists, skip lists, and their applications in sorting (bin sort, radix sort) and sparse tables.	8 hours
V	**Trees and Graphs (8 hours)** **Trees:** Rooted trees, binary search trees, spanning trees, minimal spanning trees (Kruskal's and Prim's algorithms). **Binary Trees:** Properties, traversal methods (in-order, pre-order, post-order), heaps, heap operations, heap sort, Huffman coding, and leftist trees. **Graphs:** Graph representations, traversal techniques (BFS, DFS), and applications including minimum spanning trees.	8 hours

Text Books				
Code	**Title of the Book**	**Author Name /Designation / Organization**	**Publisher**	**Edition/ Publication Year**
T1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein	MIT Press	3rd Edition, 2009.
T2	Data Structures Using C	Reema Thareja	Oxford University Press	Second Edition 2011

Reference Books				
Code	**Title of the Book**	**Author Name/Designation/ Organization**	**Publisher**	**Edition/ Publication Year**
R1	Data Structures: A Pseudo code Approach	Richard F. Gilberg and Behrouz	A. Forouzan Cengage Learning	3rd Edition, 2014.
R2	The Art of Computer Programming	Donald E. Knuth	Addison-Wesley	

Useful Links	
1	https://www.geeksforgeeks.org/data-structures/
2	https://www.coursera.org/specializations/data-structures-algorithms
3	https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/

Semester	Course Code	Name of the course	L	T	P	Credits
I	MC1T002	Object Oriented Programming using JAVA	4	-	-	4

Marks Distribution			
CA	MSE	ESE	Total
40	-	60	100

Prerequisites for the course	
1	Basic understanding of programming concepts and fundamentals
2	Basic Knowledge of Computer & Algorithms

Prior Reading Material/useful links	
1	http://www.coursera.org/projects/java-beginners-getting-started-
2	http://www.udemy.com/course/mastering-object-oriented-design-in-java/
3	https://www.udemy.com/course/oop-learnit/

Course Objectives	
Sr No	Statement
1	To introduce the basic concepts of Object-Oriented Programming (OOP).
2	To understand the fundamentals of Java and its application in internet programming.
3	To learn about Java data types, operators, and decision-making constructs.
4	To explore the concepts of classes, objects, inheritance, and polymorphism in Java.
5	To gain knowledge on handling exceptions and multithreading in Java programming.

Course Outcomes		
Sr No	Code	CO statement
1	CO1	Use the syntax and semantics of java programming language and basic concepts of OOP.
2	CO2	Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.
3	CO3	Proposed the use of certain technologies by implementing them in the Java programming language to solve the given problem
4	CO4	Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.
5	CO5	Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements

Unit No	Course Contents	Duration
I	**Basics of OOP:** Abstraction, Inheritance, Encapsulation, Classes, subclasses and super classes, Polymorphism and Overloading, message communication Procedure-Oriented vs. Object-Oriented Programming concept. **Introduction to Java Programming:** Basics of Java, Background/History of Java, Java and the Internet, Advantages of Java, Java Virtual Machine & Byte Code, Java Environment Setup, Java Program Structure	7 hours
II	**Primitive Data Types:** Integers, Floating Point type, Characters, Booleans, User Defined Data Type, Identifiers &Literals, Declarations of constants &variables, Type Conversion and Casting, Scope of variables & default values of variables declared, Wrapper classes, Comment Syntax, Garbage Collection **Arrays of Primitive Data Types:** Types of Arrays, Creation, concatenation and conversion of a string, Decision& Control Statements, Different Operators	7 hours
III	**Class:** Defining classes, fields and methods, creating objects, accessing rules, this keyword, static keyword, method overloading, final keyword **Constructors:** Default constructors, Parameterized constructors, copy constructors, Passing object as a parameter, constructor overloading	7 hours
IV	**Basics of Inheritance:** Inheritance, Types of inheritance: single, multiple, multilevel, hierarchical and hybrid inheritance, concepts of method overriding, extending class, super class, Abstract Class **Package:** Creating package, importing package, access rules for packages, class hiding rules in a package, defining interface, inheritance on interfaces, implementing interface, multiple inheritance using interface	8 hours
V	**Exception Handling:** Introduction, built in classes for Exception Handling, Mechanism of Exception Handling in Java, Error Handling Exception Classes **Multithreading:** Creating thread, extending Thread class, implementing Runnable interface, life cycle of a thread, Thread priority & thread synchronization, exception handing in threads	7 hours

Text Books				
Code	**Title of the Book**	**Author Name/Designation / Organization**	**Publisher**	**Edition/ Publication Year**
T1	"Java: The Complete Reference"	Herbert Schildt	McGraw-Hill	11th Edition/2018
T2	Effective Java	Joshua Bloch	Addison-Wesley	3rd Edition/2018

Reference Books				
Code	**Title of the Book**	**Author Name/Designation/ Organization**	**Publisher**	**Edition/ Publication Year**
R1	Head First Java	Kathy Sierra, Bert Bates	O'Reilly Media	2nd Edition/2005
R2	Java: How to Program	Paul Deitel, Harvey Deitel	Pearson	11th Edition/2017

Useful Links	
1	https://docs.oracle.com/javase/tutorial/
2	https://www.w3schools.com/java/
3	https://www.geeksforgeeks.org/java/

Semester	Course Code	Name of the course	L	T	P	Credits
I	MC1T003	Operating Systems	3			4

Marks Distribution			
CA	MSE	ESE	Total
	40	60	100

Prerequisites for the course	
1	Basic understanding of Computer Architecture
2	Knowledge of Programming in C/C++

Prior Reading Material/useful links	
1	https://www.os-book.com/
2	https://www.tutorialspoint.com/operating_system/index.htm

Course Objectives	
Sr No	Statement
1	To understand the fundamental concepts and components of operating systems.
2	To learn process management, including scheduling, synchronization, and deadlock handling.
3	To explore various memory management techniques, including paging and segmentation.
4	To study file systems and I/O management techniques.
5	To understand security, protection, and advanced OS topics like virtualization and cloud OS.

Course Outcomes		
Sr No	Code	CO statement
1	CO1	Explain the structure and functions of an operating system.
2	CO2	Apply process scheduling and synchronization techniques.
3	CO3	Analyze memory management techniques like paging and segmentation.
4	CO4	Evaluate file systems and I/O management techniques.
5	CO5	Design security mechanisms and explore advanced topics in OS.

Unit No	Course Contents	Duration
I	**Introduction to Operating Systems:** Overview, OS Structure, OS Services, System Calls, Types of OS, Batch, Multi-programmed, Time-sharing, Real-time, Distributed, Parallel and Embedded Systems.	7 hours
II	**Process Management:** Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, Threads, Multithreading Models, CPU Scheduling Algorithms, Process Synchronization, Critical Section Problem, Semaphores, Deadlocks – Avoidance, Prevention, and Recovery.	7 hours
III	**Memory Management:** Memory Management Strategies, Contiguous and Non-Contiguous Allocation, Paging, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.	7 hours
IV	**File Systems and I/O Management:** File Concept, Access Methods, Directory Structure, File System Mounting, File Sharing, Protection, Implementing File Systems, File-System Structure, Directory Implementation, Allocation Methods, Free-Space Management, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management.	8 hours
V	**Security and Advanced Topics:** Protection, Security, Authentication, Encryption, OS Security, Case Studies of Unix/Linux, Windows, Virtualization, Cloud OS Concepts, Virtual Machines.	7 hours

Text Books				
Code	**Title of the Book**	**Author Name /Designation / Organization**	**Publisher**	**Edition/ Publication Year**
T1	Operating System Concepts	Silberschatz, Galvin, Gagne	Wiley	9th Edition, 2018
T2	Modern Operating Systems	Andrew S. Tanenbaum	Pearson Education	4th Edition, 2015

Reference Books				
Code	**Title of the Book**	**Author Name/Designation/ Organization**	**Publisher**	**Edition/ Publication Year**
	Operating Systems: A Concept-Based	D.M. Dhamdhere	McGraw Hill	3rd Edition, 2012

	Approach			
	Operating Systems and Middleware	Max Hailperin	Pearson Education	1st Edition, 2006

Useful Links	
1	https://www.geeksforgeeks.org/operating-systems/
2	https://www.youtube.com/watch?v=mXw9ruZaxzQ
3	https://prepinsta.com/operating-systems/

Semester	Course Code	Name of the course	L	T	P	Credits
I	MC1T004	Discrete Mathematics & Graph Theory	3	-	-	3

Marks Distribution			
CA	MSE	ESE	Total
40	-	60	100

Prerequisites for the course	
1	Knowledge of high school level arithmetic and algebra.
2	Good understanding of elementary algebra and arithmetic.

Prior Reading Material/useful links	
1	http://www.javatpoint.com/discrete-mathematics-tutorial
2	http://www.geeksforgeeks.org/discrete-mathematics-tutorial/

Course Objectives	
Sr No	Statement
1	Introduce the basic concepts of sets, logic, functions, and relations, providing foundational knowledge for advanced topics in computer science and mathematics.
2	Equip students with the skills to apply propositional and predicate logic, along with mathematical induction, to solve mathematical problems and verify the correctness of arguments.
3	Teach the fundamental concepts of graph theory including paths, circuits, and graph representations, enabling students to solve related problems in various domains such as networking and algorithms.
4	Cover algebraic structures such as groups, monoids, and rings, and explore their properties and applications, enhancing students' understanding of abstract mathematical concepts.
5	Introduce Boolean algebra, its identities, and normal forms, emphasizing its applications in computer science, digital logic design, and simplified representations of logical expressions.

Course Outcomes		
Sr No	Code	CO statement
1	CO1	Define and illustrate fundamental set theory concepts, logical connectives, and propositional logic
2	CO2	Analyse different types of functions and relations, including injective, surjective, bijective, reflexive, symmetric, and transitive relations.
3	CO3	Solve problems related to graph theory, including identifying shortest paths, Euler and Hamiltonian paths, and graph isomorphisms
4	CO4	Evaluate algebraic structures with one or two binary operations, such as

		groups, rings, and fields, and perform operations within these structures
5	CO5	Construct and simplify Boolean functions using identities and representations such as Disjunctive and Conjunctive Normal Forms

Unit No	**Course Contents**	**Duration**
I	Fundamental Structures and Basic Logic:Sets, Venn diagram, Cartesian product, Power sets, Cardinality and countability, Propositional logic, Logical connectives, Truth tables, Normal forms, Validity, Predicate logic, Limitations of predicate logic, Universal and existential quantification, First order logic. Principles of Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.	7 hours
II	Functions and Relations: Subjective, Injective, Bijective and inverse functions, Composition of function, Reflexivity, Symmetry, Transitivity and equivalence relations	7 hours
III	Graph Theory: Basic terminology, Multi graphs and weighted graphs, Paths and circuits, shortest path problems, Euler and Hamiltonian paths, Representation of graph, Isomorphic graphs, Planar graphs, Connectivity, Matching Colouring.	7 hours
IV	Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient79 Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups	7 hours
V	Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form	8 hours

Text Books				
Code	**Title of the Book**	**Author Name/Designation / Organization**	**Publisher**	**Edition/ Publication Year**
T1	Elements of Discrete Mathematics	C. L. Liu	McGraw-Hill Publication	3rd Edition, 2008
T2	Discrete Mathematics and its Applications	Kenneth H. Rosen	McGraw-Hill Publication	8th Edition, 2019.

Reference Books				
Code	**Title of the Book**	**Author Name/Designation/ Organization**	**Publisher**	**Edition/ Publication Year**
R1	Discrete Mathematics	Lipschutz, Discrete	McGraw-Hill Publication	
R2	Discrete Mathematics with Proof	Eric Gossett	Wiley	3rd edition/2021

Useful Links	
1	https://onlinecourses.nptel.ac.in/noc20_cs82/preview
2	http://www.youtube.com/user/dmgtplc

Semester	Course Code	Name of the course	L	T	P	Credits
I	MC1A005	Computer Architecture & Organization	3	-	-	3

Marks Distribution			
CA	MSE	ESE	Total
40	-	60	100

Prerequisites for the course	
1	Basic Knowledge of Computer Hardware, Architecture, Memory etc.

Prior Reading Material/useful links	
1	Computer Organization and Architecture, 6th Edition, William Stallings
2	Computer Architecture - A Quantitative Approach,5th edition, John L. Hennessy, David A. Patterson.

Course Objectives	
Sr. No	Statement
1	Introduce the structure, function, and performance of computer systems.
2	Understand the operations of CPU, memory, and I/O devices.
3	Familiarize with assembly language and micro-level processing.
4	Develop an understanding of data representation, machine-level arithmetic, and control unit operations.
5	Impart knowledge of modern architecture trends like RISC, CISC, and pipelining.

Course Outcomes		
Sr. No	Code	CO statement
1	CO1	Understand the basic structure of computers and their functionalities.
2	CO2	Analyse the interaction between hardware and software in computing systems.
3	CO3	Demonstrate knowledge of machine instructions, addressing modes, and assembly programming.
4	CO4	Evaluate various CPU architectures and their performance metrics.
5	CO5	Explore the concepts of memory hierarchy, cache organization, and virtual memory.

Unit No	Course Contents	Duration
I	**Unit 1: Introduction to Computer Architecture** • Introduction to Computer Architecture and Organization • Basic structure of Computers: Von Neumann Architecture • Functional units of a computer system • Types of computer architectures: SISD, SIMD, MIMD, MISD • Performance metrics: Throughput, Latency, MIPS, FLOPS • Machine language and assembly language • Data representation: Number systems (Binary, Hexadecimal, Octal) • Signed number representations (1's complement, 2's complement) • Floating-point arithmetic and IEEE standards • Overview of System Bus and Bus Interconnection	6 hours
II	**Unit 2: Central Processing Unit (CPU)** • Structure and function of CPU • Arithmetic Logic Unit (ALU): Design and operation • Control Unit (CU): Hardwired vs. Microprogrammed • Instruction cycle and execution process • Types of instructions: Data transfer, Arithmetic, Logical, Control flow • Instruction formats and addressing modes • Register organization: General-purpose and special-purpose registers • Program counter, stack pointer, and status register • Interrupt handling and exception processing • Introduction to Pipelining and its types (Instruction and Arithmetic)	7 hours
III	**Unit 3: Memory System Organization** • Overview of memory hierarchy: Primary, Secondary, Cache, Virtual memory • RAM and ROM: Characteristics and types • Cache memory: Mapping techniques (Direct, Associative, Set-Associative) • Cache coherence and write policies • Memory interleaving and access optimization techniques • Virtual memory: Paging and Segmentation • Memory allocation: Dynamic, Static, and Fragmentation • RAID: Levels, benefits, and performance considerations • Memory management hardware: MMU and TLB • Introduction to modern memory technologies: DRAM, SRAM, Flash	7 hours

IV	**Unit 4: Input/Output Organization** • Introduction to I/O systems and devices • I/O addressing: Memory-mapped and isolated I/O • I/O communication techniques: Polling, Interrupt-driven, Direct Memory Access (DMA) • Types of I/O devices: Input, Output, and Storage devices • I/O data transfer techniques: Programmed I/O, Interrupt I/O, DMA • I/O interfaces: SCSI, PCI, USB • Performance measures for I/O systems • Interrupt processing and handling mechanisms • Synchronization techniques: Semaphore, Spinlocks • I/O performance improvements: Spooling, Buffering, Caching	8 hours
V	**Unit 5: Advanced Architectures and Concepts** • Introduction to Reduced Instruction Set Computing (RISC) and Complex Instruction Set Computing (CISC) • Comparison of RISC and CISC architectures • Parallel processing: Symmetric and Asymmetric Multiprocessing (SMP/AMP) • Flynn’s taxonomy and its classification of parallel architectures • Introduction to multi-core processors and their organization • Pipelining: Types, hazards (data, structural, control), and handling strategies • Introduction to superscalar and VLIW architectures • Basics of GPU architecture and its applications • Quantum computing overview and its potential impact • Trends in modern computing: Cloud, Edge, and Distributed systems	8 hours

Text Books

Code	Title of the Book	Author Name /Designation / Organization	Publisher	Edition/ Publication Year
T1	Digital Design and Computer Architecture	William Stallings	Pearson	11th Edition, 2018
T2	Computer System Architecture	M. Morris Mano	Pearson	3rd Edition, 2017

Reference Books				
Code	**Title of the Book**	**Author Name/Designation/ Organization**	**Publisher**	**Edition/ Publication Year**
R1	Advanced Computer Architecture	Kai Hwang	Tata McGraw Hill	3rd Edition, 2018
R2	Computer Organization and Embedded Systems	Carl Hamacher, Zvonko Vranesic, Safwat Zaky	McGraw Hill	6th Edition, 2014

Useful Links	
1	https://onlinecourses.nptel.ac.in/noc22_cs10/preview

Semester	Course Code	Name of the Course	L	T	P	Credits
I	MC1L006	Data Structure and Algorithm LAB	0	0	4	2

Marks Distribution			
CA	**MSE**	**ESE**	**Total**
60	0	40	100

Course Outcomes		
Sr No	**Code**	**CO statement**
1	CO1	Understand and apply fundamental concepts of data structures including arrays, pointers, and dynamic memory allocation in solving computational problems.
2	CO2	Implement and analyze the performance of linked list data structures (singly, doubly, and circular linked lists) and advanced structures like skip lists and self-adjusting lists.
3	CO3	Utilize stack and queue data structures to solve practical problems involving recursion, expression evaluation, and various queue types including priority and circular queues.
4	CO4	Develop and implement various tree structures (binary trees, AVL trees, B-trees) and graph algorithms (BFS, DFS, shortest path, spanning tree) in practical applications.
5	CO5	Analyze and apply advanced sorting, searching, and hashing techniques to optimize data handling processes, with a focus on algorithmic efficiency and performance.

List Of Experiments	
Sr No	**Description**
1	Implement basic operations on arrays: insertion, deletion, updating, and searching elements.
2	Write a C++ program to implement bubble sort and display the sorted array**.**
3	Implement a singly linked list with basic operations: insertion, deletion, and traversal.
4	Develop a program to implement doubly linked lists and circular linked lists, including insertion and deletion operations.
5	Develop a C++ program to create a queue using linked lists and perform enqueue, dequeue, and display operations.
6	Implement standard queue operations and a circular queue using arrays.
7	Implement a stack using arrays in C++, and perform push, pop, and display operations.
8	Implement sorting algorithms such as quick sort, merge sort, and heap sort, and compare their performance.
9	Create a binary search tree (BST**)** in C++ and perform insert, delete, and search operations.
10	Implement Depth First Search (DFS) algorithm for a graph using adjacency matrix representation in C++.

<table>
<tr><th colspan="2">Content Beyond Syllabus</th></tr>
<tr><th>Sr No</th><th>Description</th></tr>
<tr><td>1</td><td>Introduction to implementing <b>AVL trees</b> in C++ and understanding their self-balancing properties.</td></tr>
<tr><td>2</td><td>Explore <b>hashing</b> techniques and implement basic hash functions in C++..</td></tr>
</table>

Semester	Course Code	Name of the Course	L	T	P	Credits
I	MC1L007	Object Oriented Programming using JAVA Lab	-	-	4	2

Marks Distribution			
CA	MSE	ESE	Total
60	-	40	100

Course Outcomes		
Sr No	Code	CO statement
1	CO1	Explain the fundamental concepts of Object-Oriented Programming including classes, objects, inheritance, polymorphism, and encapsulation.
2	CO2	Apply object-oriented programming principles to design and implement Java programs using classes, objects, and methods.
3	CO3	Demonstrate the use of inheritance and polymorphism to create reusable and modular Java code.
4	CO4	Analyze and handle exceptions effectively in Java programs to ensure robust and error-free applications.
5	CO5	Create packages and interfaces to organize Java programs and implement multiple inheritances through interfaces.

List Of Experiments	
Sr No	Description
1	Write a simple Java program to display "Hello, World!". Learn the basics of Java environment setup, compilation, and execution process.
2	Implement a Java program that defines a class with fields and methods, creates objects, and demonstrates accessing methods using objects.
3	Create a program to demonstrate encapsulation by defining private fields and accessing them through public getter and setter methods.
4	Write a Java program to demonstrate constructor overloading by creating multiple constructors with different parameters.
5	Implement a Java program to show single inheritance. Define a base class and a derived class, and demonstrate the use of the `super` keyword.
6	Develop a program to show polymorphism by implementing method overloading and method overriding concepts in Java..
7	Write a Java program that defines an abstract class and an interface. Implement these in a subclass and demonstrate method implementation.
8	Create a Java program to handle exceptions using try-catch blocks, demonstrating the use of multiple catch blocks and finally block.
9	Write a Java program that demonstrates array operations such as initialization, accessing elements, and performing basic array manipulations.
10	Develop a program that creates a package, defines classes within the package, imports the package, and demonstrates access rules for the package.

Content Beyond Syllabus	
Sr No	Description
1	**Introduction to JavaFX for GUI Development:** Basics of JavaFX, a framework for building rich client applications, and creating a simple GUI application.

2	**Lambda Expressions and Functional Interfaces:** Introduction to lambda expressions and their use in Java 8 to create concise and flexible code blocks.

Semester	Course Code	Name of the Course	L	T	P	Credits
II	MC1L008	Web Development LAB	-	-	4	2

Marks Distribution			
CA	MSE	ESE	Total
60	-	40	100

Course Outcomes		
Sr No	Code	CO statement
1	CO1	Understand and apply HTML5 to create structured and semantic web pages.
2	CO2	Utilize CSS for styling and implementing responsive design in web pages.
3	CO3	Develop dynamic and interactive web applications using JavaScript.
4	CO4	Apply advanced JavaScript techniques including ES6, Promises, and AJAX.
5	CO5	Integrate modern front-end tools like Bootstrap, jQuery, and React.js.

List Of Experiments	
Sr No	Description
1	**Basic Web Page Layout (HTML + CSS)**: Create a simple web page with headings, paragraphs, lists, and images.
2	**Form Creation (HTML + CSS)**: Design a contact form with input fields (name, email, message) and a submit button.
3	**Styling with External CSS (CSS)**: Build a webpage and link an external CSS file for styling various elements (e.g., fonts, colors, margins).
4	**Responsive Grid Layout (Bootstrap)**: Use Bootstrap's grid system to create a responsive layout with multiple columns.
5	**Navigation Bar (Bootstrap)**: Implement a responsive navigation bar with dropdowns using Bootstrap.
6	**Image Gallery (HTML + CSS)**: Create an image gallery with grid layout, and add hover effects using CSS.
7	**Interactive Button (JS + CSS)**: Add a JavaScript-powered button that changes color or text when clicked.
8	**Form Validation (JavaScript)**: Implement real-time form validation using JavaScript (e.g., check for empty fields or valid email).
9	**Modal Popup (Bootstrap + JS)**: Use Bootstrap's modal component to create a popup that appears when a button is clicked.
10	**Slide Show (HTML + JS)**: Build a simple image slider that automatically cycles through images using JavaScript.

Content Beyond Syllabus	
Sr No	Description
1	**Responsive Web Design Principles:** Study advanced responsive design concepts like media queries, viewport settings, and fluid layouts.
2	**SEO Basics for Web Development:** Learn how to optimize HTML structure, metadata, and content for better search engine rankings.