Syllabus of I Semester M.Sc. Programme

Choice Based Credit System (CBCS)

(According to new regulations w.e.f. 2020-21)

Semester-I	I SEMSTER M.Sc. w.e.f. 2024-25								
	Course	Subject Name	Teaching Hrs./ week	Practical Hrs./ week	Examination				Credits
					Duration (Hrs.)	Marks			
						Theory/ Practical	IA	Total	
Core Subject	24MSCCS 1.1	Discrete Mathematical Structures	4	--	3	80	20	100	4
	24MSCCS 1.2	Web Programming	4	--	3	80	20	100	4
	24MSCCS 1.3	Object Oriented Programming (OOPS) using C++	4	--	3	80	20	100	4
	24MSCCS PL 1.4	Web Programming Lab	--	4	3	80	20	100	4
	24MSCCS PL 1.5	Object Oriented Programming (OOPS) using C++(Lab)	--	4	3	80	20	100	4
Soft Core / Specialization/ Optional	24MSCSC 1.6	Computer System Architecture	4	--	3	80	20	100	4
		Total	16	8				600	24

CS: Core Subject SC: Soft Core PL: Practical

Year	I	**Course Code:** 24MSCCS 1.1	**Credits**	04
Semester	I	**Course Title:** DISCRETE MATHEMATICAL STRUCTURES	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***The objective of this course is to emphasizes the goal of*** o Essential understanding of sets, functions, relations, and logic across disciplines. o Utilize mathematical reasoning and proofs to solve discrete structure challenges. o Assess algorithm correctness, efficiency, and complexity using discrete mathematics. o Apply graph theory, combinatorics, cryptography, and data structures in practice. Master induction, contradiction, and proofs for precise reasoning in discrete structures.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Understand fundamental concepts in sets, logic, proofs, graph theory, trees, and algebraic structures. o Formulate and analyze propositions, arguments, and proofs using logical equivalence and inference rules. o Gain proficiency in graph theory, including paths, cycles, and graph isomorphism. o Learn about tree structures and algorithms for spanning and shortest paths. o Study algebraic structures like semi groups and groups, focusing on their applications in coding and error detection.

Unit No.	Course Content	Hours
UNIT I	**Logic:** Introduction, propositional logic, propositional equivalences, predicates and quantifiers, rules of inference. **Proofs:** Introduction to proofs, proof methods.	10
UNIT II	**Sets, Functions and Relations:** Sets, set operations, functions, relations, equivalence relations and partial ordering. **Counting:** Basics of counting, the pigeonhole principle, permutations, and combinations, Binomial Co-efficient, recurrence relations.	10
UNIT III	**Elementary Combinatory:** Basis of counting, Combinations & Permutations, With repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, The principles of Inclusion – Exclusion, Pigeon- hole principles and its applications.	12
UNIT IV	**Graph Theory:** Introduction of Graphs and digraphs, Paths and Cycles, Hamiltonian Cycles, adjacency and incidence matrices, vertex coloring, representations of graphs, isomorphism's of graphs, planar graphs.	10
UNIT V	**Trees:** Terminology and characterizations of trees, spanning trees, minimal spanning trees, shortest-path algorithm, binary trees, tree traversals, decision trees, isomorphism of trees.	10

Recommended Learning Resources

Textbooks:

1. "Discrete Mathematics and its Applications, 5/e" by, Kenneth H. Rosen, Tata McGraw Hill.
2. Deo N., Graph theory with application to Engineering and Computer Science, Prentice Hall of India,
3. Kolman, Busby, Ross, Discrete Mathematical Structures, Pearson Education.

Reference Books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical structures with applications to Computer Science, Tata Hill

<table>
<tr><td><b>Year</b></td><td>I</td><td colspan="3"><b>Course Code:</b> 24MSCCS 1.2</td><td><b>Credits</b></td><td>04</td></tr>
<tr><td><b>Semester</b></td><td>I</td><td colspan="3"><b>Course Title:</b> WEB PROGRAMMING</td><td><b>Hours</b></td><td>52</td></tr>
<tr><td colspan="3">Course Pre-requisites if any</td><td colspan="4">Fundamentals on Programming</td></tr>
<tr><td colspan="3">Formative Assessment Marks: 20</td><td colspan="2">Summative Assessment Marks: 80</td><td colspan="2">Duration of ESA: 03hrs.</td></tr>
<tr><td>Course Objective</td><td colspan="6"><i>The objective of this course is to emphasizes the goal of</i>
o Understand the basic principles of web development including HTML, CSS, and JavaScript
o Learn server-side scripting languages like PHP, Python (Django/Flask), or Node.js to create dynamic web applications.
o Integrate databases (e.g., MySQL, PostgreSQL) with web applications for data storage and retrieval.
o Explore front-end frameworks (e.g., React, Angular, Vue.js) and back-end frameworks (e.g., Express.js, Django, Flask) to streamline development and enhance functionality.
o Implement security best practices (e.g., XSS prevention, CSRF protection) and optimize web applications for performance (e.g., caching, asynchronous loading).</td></tr>
<tr><td>Course Outcomes</td><td colspan="6"><i>After completing this course satisfactorily, a student will be able to:</i>
o Gain foundational knowledge of HTML and XHTML, including their structure, version history, and document types.
o Understand the advancements and features of HTML5, including document structure changes, semantic enhancements, and support for multimedia and graphics.
o Learn the basics of CSS, including selectors, syntax, cascading, and various properties for styling elements.
o Develop skills in using CSS for table formatting, web accessibility, responsive images, and navigation within web pages.
o Acquire a working knowledge of JavaScript, focusing on functions, variables, the Document Object Model (DOM), and client-side form processing.</td></tr>
<tr><td>Unit No.</td><td colspan="5">Course Content</td><td>Hours</td></tr>
<tr><td>UNIT I</td><td colspan="5"><b>Traditional HTML and XHTML:</b> First Look at HTML and XHTML, Hello HTML and XHTML World. <b>HTML and XHTML:</b> Version History, HTML and XHTML <b>DTDs:</b> The Specifications Up Close, (X) HTML Document Structure, Browsers and (X)HTML, The Rules of (X)HTML, Major Themes of (X)HTML, The Future of Mark-up—Two Paths.</td><td>12</td></tr>
<tr><td>UNIT II</td><td colspan="5">Hello HTML5, Loose Syntax Returns, XHTML5, <b>HTML5:</b> Embracing the Reality of Web Mark-up, Presentational Mark-up Removed and Redefined, HTML5 Document Structure Changes, Adding Semantics, HTML5's Open Media Effort, Client-Side Graphics with <canvas>, HTML5 Form Changes, Emerging Elements and Attributes to Support Web Applications.</td><td>12</td></tr>
</table>

UNIT III	**Introduction:** CSS Overview, CSS Rules, Example with Type Selectors and the Universal Selector, CSS Syntax and Style, Class Selectors, ID Selectors, span and div Elements, Cascading, style Attribute, style Container, External CSS Files, CSS Properties, Colour Properties, RGB Values for Colour, Opacity Values for Colour, HSL and HSLA Values for Colour, Font Properties, line-height Property, Text Properties, Border Properties, Element Box, padding Property, margin Property.	10
UNIT IV	**Table Elements, Formatting a Data Table:** Borders, Alignment, and Padding, CSS Structural Pseudo Class Selectors, thread and tbody Elements, Cell Spanning, Web Accessibility, CSS display Property with Table Values, an Element, Relative URLs, Navigation Within a Web Page, CSS for Links, **Bitmap Image Formats:** GIF, JPEG, PNG, image Element, Responsive Images, Positioning Images, Shortcut Icon, iframe Element.	10
UNIT V	History of JavaScript, Hello World Web Page, Buttons, Functions, Variables, Identifiers, Assignment Statements and Objects, Document Object Model, Forms and How They're Processed: Client-Side Versus Server-Side, form Element, Controls, Text Control, accessing a Form's Control Values, reset and focus Methods	08

Recommended Learning Resources

Textbooks:

1. HTML & CSS: The Complete Reference Thomas A. Powell, Fifth Edition, Tata McGraw Hill,
2. TextBook-2: WEB PROGRAMMING with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, First Edition

Reference Books:

1. Video Links: https://onlinecourses.swayam2.ac.in/aic20_sp11/preview

Year	I	**Course Code:** 24MSCCS 1.3	**Credits**	04
Semester	I	**Course Title:** OBJECT ORIENTED PROGRAMMING WITH C++	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		
Course Objective	***The objective of this course is to emphasizes the goal of*** o Master principles like encapsulation, inheritance, and polymorphism, essential for OOP. o Develop fluency in C++ syntax, features, and standard library usage. o Design and implement robust object-oriented solutions to real-world problems. o Understand and apply memory management techniques like pointers, dynamic memory allocation. o Adopt best practices for code reuse, maintainability, and scalability in C++.			
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Develop a strong foundation in C++ programming, including understanding data types, variables, and object-oriented programming (OOP) concepts. o Gain proficiency in defining and using classes, including constructors, destructors, and advanced class features such as friend functions and operator overloading. o Understand and implement various inheritance models and polymorphism in C++, including virtual functions and abstract classes. o Master exception handling techniques in C++ and effectively use the I/O Stream Library for file and stream operations. o Utilize the Standard Template Library (STL) components, such as containers and iterators, to efficiently manage and manipulate data structures.			
Unit No.	**Course Content**	**Hours**		
UNIT I	**Introduction to C++:** OOPS Concepts, The C++ Data Types, Literal constants, Integers, Floating point constants, The bool data type, Character constants, String constant, Escape sequences, Variables, Declaration of Variables, The Name of a Variables, C++ Keywords, The Definition of an Object, Pointer Types, String Types, Const Qualifier, Reference Types, Enumeration Types, Array Types, Multidimensional Arrays, Relationship of Array and Pointer Types, Typedef Names, Volatile Qualifier, Class Types.	10		
UNIT II	**Classes:** Class definition, Data members, Member functions, Member access, Friend functions, Class declaration versus class definition, Class object sates, Class member functions, The implicit this pointer, Static class members, Pointer to class member, Nested classes, Class initialization, The class constructor, The class Destructor, Class object arrays, Member-wise Initialization, Member-wise assignment, Function overloading, Operator overloading.	10		

UNIT III	**Inheritance:** Definition, Defining the derived class, Types of derivation, Single inheritance, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract Classes, Inheritance and constructors, Inheritance and Destructors. **Polymorphism and Virtual functions:** Polymorphism, Pointers to objects, Pointers to Derived classes, Virtual functions, Virtual destructors.	12
UNIT IV	**Exception Handling:** C-style exception handling, Terminating the program, Checking the parameters before the function call statement, C++ style exception handling, unwinding of the stack, need to throw class objects, Accessing the thrown object in the catch block. **I/O Stream Library:** Streams, Stream class hierarchy, Unformatted input-output, Formatted input-output, File Input and Output.	10
UNIT V	**Standard Template Library (STL):** Standard Template Library, Components of STL, Containers, Sequence container, Associative container, Derived container, the list class, Inserting the objects into the list, Deleting the objects from the list, Traversing the list, other operations on list, The set class, The vector class, The multiset class, The map class, The multimap Class.	10

Recommended Learning Resources

Textbooks:

1. The C++ Programming Language, Fourth Edition - Bjarne Stroustrup
2. C++ Primer, Fourth Edition By Stanley B. Lippman, Josée Lajoie, Barbara E. Moo

Reference Books:

1. A Complete Guide to Programming in C++ - Peter Prinz, Ulla Prinz
2. STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library (Addison-Wesley Professional Computing Series)

Year	I	**Course Code:** 24MSCPL 1.4	**Credits**	03
Semester	I	**Course Title:** WEB PROGRAMMING LAB	**Hours**	04 Hr/lab
Course Pre-requisites if any	Knowledge of Web Programming Language			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		
Course Objective	*The objective of this course is to emphasizes the goal of* ○ Practice implementing static web pages using HTML for structure and CSS for styling. ○ Develop interactive web pages using JavaScript for client-side scripting, event handling, and DOM manipulation. ○ Implement server-side scripting using PHP, Python (Django/Flask), or Node.js to handle form submissions, process data, and interact with database ○ Explore and integrate front-end frameworks (e.g., React, Angular, Vue.js) and back-end frameworks (e.g., Express.js, Django, Flask) to build dynamic and responsive web applications. ○ Implement security measures such as input validation and protection against common web vulnerabilities (e.g., XSS, CSRF). Optimize web application performance through caching, asynchronous loading, and resource compression.			
Course Outcomes	*After completing this course satisfactorily, a student will be able to:* ○ Analyze a web page and identify its core elements and attributes. ○ Create static web pages using HTML and CSS following best practices. ○ Implement basic interactivity using JavaScript (client-side scripting). ○ Understand and utilize web development tools and technologies effectively.			

PART A: HTML and CSS

1. Write an HTML program to create a simple webpage with a heading, paragraph of text, and an image. Use appropriate HTML tags to structure the content.
2. Create an HTML page with two paragraphs. Use inline CSS styles within the paragraph tags to change the font color and size of each paragraph.
3. Write an HTML program with a link to another webpage. Include the necessary attributes in the <a> tag to specify the target URL and display text for the link.
4. Develop a simple HTML table with two rows and three columns. Fill the table cells with some sample data (text or numbers).
5. Create an HTML page with an image. Use CSS properties like text-align or float to position the image in the center of the webpage.

PART B: JAVA SCRIPT

6. Write a simple HTML page with a button. Use JavaScript to display an alert message with a greeting when the button is clicked.
7. Create an HTML page with a paragraph element. Write JavaScript code to change the text color of the paragraph to red when the mouse hovers over the paragraph (use onmouseover event).
8. Develop a JavaScript program that prompts the user for a number using prompt(). Use an if-else statement to check if the number is even or odd and display an appropriate message using alert ().
9. Write a JavaScript function that takes two numbers as arguments. The function should return the sum of the two numbers.
10. Create an HTML page with a paragraph element. Use JavaScript to get the current date and display it within the paragraph element.

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test, 20 marks shall be awarded.

Year	I	**Course Code:** 24MCACS PL 1.5	**Credits**	03
Semester	I	**Course Title:** Object Oriented Programming with C++ Lab	**Hours**	04 Hr/lab

Course Pre-requisites if any	Knowledge of C++ Programming Language	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	*The objective of this course is to emphasizes the goal of* ○ Implement classes, inheritance, polymorphism, and encapsulation in C++ programs. ○ Develop applications using C++ to solve practical programming challenges. ○ Apply debugging techniques and perform systematic testing of C++ programs. ○ Understand and apply design patterns such as singleton, factory, and observer patterns. ○ Collaborate on larger projects to integrate learned concepts and showcase practical programming skills.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Understand the difference between object-oriented programming (OOP) and procedural programming. ○ Apply C++ features such as classes, objects, constructors, destructors, inheritance, and templates in program design and implementation. ○ Analyze the characteristics of OOP and build object-oriented software using C++. ○ Assess OOP features like virtual functions, polymorphism, and exception handling in comparison to other programming languages.

PART A

1. Write a C++ program to find roots of quadratic Equation
2. Program to do banking operations using constructor and destructor functions
3. Program to find area of cone , square and triangle using inline member function
4. Program to find factorial of a given number using friend function
5. Write a C++ program to demonstrate single inheritance of library transactions
6. Write a C++ program to demonstrate multiple inheritance on students class
7. Write a C++ program to addition and subtraction of complex numbers using operator overloading
8. Program to demonstrate polymorphism concept.
9. Program to sort integers and strings using function templates

PART B:

10. Program to create and append records into employee data file
11. Write a program to search an element of array using binary search method
12. Write a program sort elements of array using Selection sort method
13. Write a C++ program to do STACK operations
14. Write a C++ program to implement evaluation of Expression
15. Program to do Queue operations
16. Write a C++ program to do operations of Circular Queue
17. Write a C++ program to demonstrate Singly Liked List
18. Write a C++ program to Demonstrate Double Liked List.

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test of 20 marks shall be awarded.

Year	I	**Course Code:** 24MSCSC 1.6	**Credits**	04
Semester	I	**Course Title:** COMPUTER SYSTEM ARCHITECTURE	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** o Understand computer data representation, including various data types, complements, and both fixed and floating point representations. o Explore basic computer organization and design, focusing on instruction codes, registers, memory-reference instructions, and the instruction cycle. o Gain proficiency in programming a basic computer using machine and assembly languages, with emphasis on arithmetic and logic operations, subroutines, and I/O programming. o Study the design and functionality of the Central Processing Unit (CPU), including general register and stack organizations, instruction formats, and addressing modes. o Analyze pipeline and vector processing techniques, covering Flynn's taxonomy, parallel processing, and RISC pipeline architectures.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Understand the various types of number systems (e.g., binary, decimal) and their conversions. o Boolean expressions using minimization methods (such as Karnaugh maps and Boolean theorems) through logical gates. o Design both combinational and sequential circuits. o Learn state reduction techniques to solve sequential circuits. o Foundational knowledge about different types of memories used in digital systems.

Unit No.	Course Content	Hours
UNIT I	Computer Data Representation- Basic computer data types, Complements, Fixed point representation, Register Transfer and Micro-operations: Floating point representation, Register Transfer language, Register Transfer, Bus and Memory Transfers (Tree-State Bus Buffers, Memory Transfer), Arithmetic Micro-Operations, Logic Micro Operations, Shift Micro-Operations, Arithmetic logical shift unit. Basic Computer Organization and Design -Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions	10

UNIT II	Basic Computer Organization and Design Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input output and interrupt, Complete computer description, Design of Basic computer, design of Accumulator Unit.	10
UNIT III	Programming The Basic Computer Introduction, Machine Language, Assembly Language, assembler, Program loops, Programming Arithmetic and logic operations, subroutines, I-O Programming. Micro programmed Control: Control Memory, Address sequencing, Micro program Example, design of control Unit.	12
UNIT IV	Central Processing Unit Introduction, General Register Organization, Stack Organization, Instruction format, Addressing Modes, data transfer and manipulation, Program Control, Reduced Instruction Set Computer (RISC).	10
UNIT V	Pipeline And Vector Processing Flynn's taxonomy, Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction, Pipeline, RISC Pipeline, Vector Processing, Array Processors.	10

Recommended Learning Resources

Textbooks:

1. M. Morris Mano, Computer System Architecture, Pearson publications.
2. Andrew S. Tanenbaum and Todd Austin, Structured Computer Organization, Sixth Edition, PHI

Reference Books:

1. Zvi. Kohavi (2004), Switching and Finite Automata Theory, Tata McGraw Hill, India.
2. C. V. S. Rao (2009), Switching and Logic Design, 3rd Edition, Pearson Education, India.
3. Donald D. Givone (2002), Digital Principles and Design, Tata McGraw Hill, India
4. Roth (2004), Fundamentals of Logic Design, 5th Edition, Thomson, IndiaLet Us C by Yashwanth Kanethar.
5. “Programming in ANSI C” by E. Balaguruswamy.
6. Complete Reference of C++ by Herbert Schilde.
7. Rajender Singh Chhillar: Application of IT to Business, Ramesh Publishers, Jaipur.
8. Gill Nasib Singh: Computing Fundamentals and Programming in C, Khanna Books, Publishing Co., New Delhi.

Syllabus of II Semester M.Sc. Programme

Choice Based Credit System (CBCS)

(According to new regulations w.e.f.2020-21)

Semester-II	II SEMSTER M.Sc. w.e.f. 2024-25								
	Course	Subject Name	Teaching Hrs./ week	Practical Hrs./ week	Examination				Credits
					Duration (Hrs.)	Marks			
						Theory/ Practical	IA	Total	
Core Subject	24MSCCS 2.1	Java Programming	4	--	3	80	20	100	4
	24MSCCS 2.2	Data Structures and Algorithms	4	--	3	80	20	100	4
	24MSCCSPL 2.3	Java Programming Lab	--	4	3	80	20	100	4
	24MSCCSPL 2.4	Data Structures and Algorithms Lab	--	4	3	80	20	100	4
Soft Core / Specialization / Optional	24MSCSC 2.5	Data Communication and Network Security	4	--	3	80	20	100	4
Note: Students has to Choose any one Soft Core / Specialization/ Optional									
Open Elective	24MCAOE 2.6	a. Computer Fundamentals and Its Applications b. Internet Concepts and Web Design	4	--	3	80	20	100	4
	Total		20	8				600	24

CS: Core Subject SC: Soft Core PL: Practical Lab OE: Open Elective

Year	I	Course Code: 24MSCCS 2.1	Credits	04
Semester	II	Course Title: PROGRAMMING USING JAVA	Hours	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	*The objective of this course is to emphasizes the goal of* ○ Develop proficiency in Java syntax, object-oriented programming principles, and core libraries. ○ Build and deploy Java applications for various platforms, including desktop, web, and mobile. ○ Implement and utilize data structures and algorithms in Java for efficient problem-solving. ○ Understand and apply Java's concurrency features for developing multithreaded applications. ○ Learn to create graphical user interfaces (GUIs) using Java's Swing or JavaFX libraries for interactive applications.	
Course Outcomes	*After completing this course satisfactorily, a student will be able to:* ○ Develop proficiency in Java programming fundamentals, including object-oriented concepts, inheritance, exception handling, and interfaces, enabling students to build robust applications. ○ Acquire practical skills in input/output operations, multithreading, and utilizing Java collections for efficient data manipulation. ○ Master GUI development using AWT and Swing, event handling, and servlet programming for creating interactive and dynamic web applications. ○ Gain expertise in database connectivity and client-server communication through JDBC, RMI, and networking concepts, facilitating the development of database-driven applications. ○ Understand Java Server Pages (JSP) basics, directive elements, and custom tags, along with MVC architecture, empowering students to develop scalable and maintainable web applications.	
Unit No.	Course Content	Hours
UNIT I	**FUNDAMENTALS OF JAVA PROGRAMMING:** Review, Class and Objects, Inheritance in Java, Inheritance in classes, Using super - Method overriding, Dynamic Method Dispatch, Abstract Classes, Using final with inheritance, the Object Class, Interfaces and Packages, Inheritance in java with Interfaces, Defining Interfaces, Implementing Interfaces, Extending Interfaces, Creating Packages, CLASSPATH variable, Access protection, Importing Packages, Interfaces in a Package, Exception Handling in Java, try-catch-finally mechanism - throw statement - throws statement - Classes for Exception Handling.	10

UNIT II	**INPUT / OUTPUT IN JAVA, MULTI THREADING, APPLETS:** Input / Output in java - java.io package, I/O Streams - Readers and Writers - Using various I/O classes – Reader, Writer, Input Stream, and Output Stream. **Serialization of objects Multithreading:** Life cycle of a thread - Java Thread priorities - Runnable interface and Thread Class - Sharing limited Resources - Shared Object with Synchronization, Comparators, Collections, Collection-classes, List – Set – Maps – Trees – Iterators.	12
UNIT III	**GUI COMPONENTS (AWT& SWING), SWING, SERVLETS:** GUI concepts in java, Basic GUI Components in AWT, Container Classes, Layout Managers, Flow Layout, Border Layout-Card Layout-Box Layout, Difference between AWT and SWING, Event Handling, Handling Keyboard Events and Mouse Events, Handling Sessions and Cookies, Servlet Model – Overview, Environment Setup: Life Cycle, Examples, Client Request - Server Response.	10
UNIT IV	**DATABASE AND CLIENT SERVER COMMUNICATION:** Networking - Creating a server that sends data, Creating a client that receives data, two way communications between server and client, Difference between Server Socket and Socket, RMI, JDBC, Using MS-SQL Server Stages in a JDBC program, Registering the driver, Connecting to database, Transaction and Non-Transactional Events, Preparing SQL statements, various methods of statements and differences, Improving the performance of a JDBC program.	10
UNIT V	**JSP BASICS, DIRECTIVE ELEMENTS, CUSTOM TAGS:** Java Server Pages, The Problem with Servlets, Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP, Environment: JSP Directives, JSP Action, JSP Implicit Objects, JSP Form Processing, JSP Session and Cookies Handling - JSP Session. Tracking - JSP Database Access - JSP Standard Tag Libraries - JSP Custom Tag - JSP Expression Language - JSP Exception Handling - JSP XML Processing.	10

Recommended Learning Resources

Textbooks:

1. Schildt Herbert, “The Complete Reference”, Java Eighth Edition, Tata McGraw-Hill,2011
2. Kathy walrath, “Java server programming J2EE”, 1st ed., Black Book, Dream Tech Publishers,2015

Reference Books:

1. Deitel & Deitel, “Java How to Program”, Pearson Education Asia, 10th Edition, 2015.
2. Rao Nageswara, “Core Java: An Integrated Approach”, Dream tech press, 2nd Edition, 2010.
3. James Keogh, “Complete Reference J2EE” McGraw publication, 2015.

Year	I	**Course Code:** 24MSCCS 2.2	**Credits**	04
Semester	II	**Course Title:** DATA STRUCTURES AND ALGORITHMS	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		
Course Objective	***The objective of this course is to emphasizes the goal of*** o Master fundamental data structures like arrays, linked lists, trees, and graphs. o Analyze algorithms for time and space complexity using Big-O notation. o Implement and compare various sorting and searching algorithms. o Explore advanced structures such as heaps, hash tables, and balanced trees. o Apply data structures and algorithms to solve real-world computational problems efficiently.			
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Understand fundamental concepts of computation, algorithms, and data structures, including the implementation and analysis of iterative and recursive algorithms. o Master sorting algorithms such as selection sort, insertion sort, quicksort, and merge sort, with a focus on their correctness and efficiency analysis. o Gain proficiency in asymptotic analysis, including the use of asymptotic notation, solving recurrences, and applying these techniques to evaluate algorithm efficiency. o Develop skills in implementing and using linear data structures like arrays and linked lists, and their applications in stack and queue operations. o Learn about advanced data structures, including various types of trees and their applications, as well as graph algorithms and design paradigms like greedy, divide and conquer, dynamic programming, and backtracking.			

Unit No.	Course Content	Hours
UNIT I	**Introduction:** Computation, algorithms, and data structures. Iteration and Recursion, Iterative and recursive algorithms (Towers of Hanoi, Euclid's GCD algorithm), proof of correctness using induction. Sorting arrays using selection sort, insertion sort, quicksort, and merge sort (with proof of correctness and efficiency analysis).	10
UNIT II	**Asymptotic Analysis:** Asymptotic notation. Estimates of binomials and factorials. Methods of solving recurrences using induction, recursion tree and Master method. Application of asymptotic to analyzing efficiency of algorithms.	12
UNIT III	**Linear Data Structures:** Arrays and linked lists (single and doubly linked lists). Array and linked list-based implementations for Queue and Stack. Applications to infix-postfix conversion and expression evaluation.	10

UNIT IV	**ADT Tree:** tree representation, traversal of trees; ADT Binary tree - binary trees, threaded binary trees, application of binary trees - Huffmann coding; application of threaded binary trees - differentiation; Search Tree - Binary search tree; balanced binary search trees - AVL tree; Applications of Search Trees - TRIE; 2-3 tree, 2-3-4 tree; concept of B-Tree.	10
UNIT V	**Graphs:** Graphs - shortest path, minimum spanning tree, DFS, BFS - an application of DFS and BFS. Algorithm Design Paradigms - greedy, divide and conquer, dynamic programming, backtracking. Adjacency matrix and adjacency list representations.	10

Recommended Learning Resources

Textbooks:

1. Introduction to Algorithms, by T. H. cormen, C.E. Leiserson, R. L. Rivest, C. Stein
2. Algorithms in C by Robert Sedgewick Algorithm design and Applications by M.T. Goodrich, R.Tamassia.

Reference Books:

1. E. Horowitz & Sahni,Fundamental Data Structure, Galgotia Book Source, 1983.
2. A. Tannenbaum, Data Structure Using C, Pearson Education, 2003.
3. Kruz, Data Structure and Programming Design, 1987.
4. N. Wirth, Algorithms +Data Structure = Program, Prentice Hall of India, 1979.
5. Goodrich & Tamassia, Data Structures and Algorithms in C++, 2nd Edition, John Wiley & Sons, 2011.

Year	I	**Course Code:** 24MSCCSPL 2.3	**Credits**	03
Semester	II	**Course Title:** JAVA PROGRAMMING LAB	**Hours**	04 Hr/lab
Course Pre-requisites if any	N/A			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		
Course Objective	*The objective of this course is to emphasizes the goal of* o Develop proficiency in writing Java programs to solve various programming challenges. o Gain hands-on experience in developing Java applications, including desktop, web, and mobile applications. o Learn debugging techniques and perform systematic testing of Java programs for correctness and efficiency. o Apply object-oriented design principles to create well-structured Java programs with reusable components. o Practice integrating Java APIs and libraries to extend functionality and enhance application development.			
Course Outcomes	*After completing this course satisfactorily, a student will be able to:* o Implement core Java syntax, control flow, and object-oriented concepts. o Debug and test Java programs effectively using tools and test cases. o Design classes, objects, and utilize inheritance for code reusability. o Solve problems using Java code, basic data structures, and analyze efficiency. o Write basic Java programs that interact with user input and output.			

Practice Lab programs:

1. Implement a basic calculator that performs addition, subtraction, multiplication, and division on user-provided numbers.
2. Create a program that prompts for the dimensions (length, width, or radius) and calculates the area and perimeter (or circumference) of shapes like rectangles, squares, and circles.
3. Write a program that converts between different units (e.g., Celsius to Fahrenheit, kilometers to miles).
4. Develop a program that determines if a given year is a leap year based on divisibility rules.
5. Write programs to check for specific number patterns like Armstrong numbers (sum of digits' cubes equals the number) or palindrome strings (reads the same backward as forward).
6. Create a program that demonstrates string manipulation methods like finding length, extracting characters, comparing strings, and converting case.
7. Develop a program that showcases basic array functionalities like initialization, element access, calculations on elements (e.g., sum, average), and iterating through the array using loops.
8. Design a program that presents a menu with options like the above basic functionalities (calculator, converter) and allows the user to choose and execute the desired program.
9. Develop a program that calculates total marks, percentage, and grade based on user-provided subject marks, using predefined criteria and displaying the results.
10. Implement basic file operations like reading text content from a file and writing simple data (e.g., user input) to a text file.

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test, 20 marks shall be awarded.

Year	I	**Course Code:** 24MSCCSPL 2.4	**Credits**	03
Semester	II	**Course Title:** DATA STRUCTURES AND ALGORITHMS LAB	**Hours**	04 Hr./lab
Course Pre-requisites if any	N/A			
Formative Assessment Marks: 20	Summative Assessment Marks: 80		Duration of ESA: 03hrs.	
Course Objective	*The objective of this course is to emphasizes the goal of* ○ Implement fundamental data structures like arrays, linked lists, stacks, queues, and trees using C++. ○ Implement various algorithms such as sorting, searching, graph traversal, and dynamic programming in C++. ○ Analyse the time and space complexity of implemented algorithms using Big-O notation. ○ Solve programming problems using appropriate data structures and algorithms efficiently. ○ Collaborate on projects that integrate data structures and algorithms to solve real-world computational problems effectively.			
Course Outcomes	*After completing this course satisfactorily, a student will be able to:* ○ Implement core data structures (arrays, linked lists, stacks, queues) in a chosen programming language. ○ Design and analyze algorithms for efficiency (Big O notation). ○ Solve problems using DSA concepts and efficient code. ○ Debug and test DSA implementations effectively. ○ Build a foundation for exploring advanced data structures and algorithms.			

PART A: Linear Data structure

1. Write a C++ program to implement the Stack ADT using an Array.
2. Write a C++ program to evaluate postfix evaluation.
3. Write a C++ program to convert an expression from Infix to Postfix.
4. Write a C++ program to implement using recursive functions.
   a. Linear Search
   b. Binary Search
5. Write a C++ program to implement using non-recursive functions.
   a. Linear Search
   b. Binary Search
6. Write a C++ program of Tower of Hanoi.
7. Write a C++ program to implement the Queue ADT using an Array.
8. Write a C++ program to implement the Linked List editing to perform following operations.
   a. Insert an element into a list, Delete an element from the list, Search for a key element in the list and Count number of nodes in the list.
9. Write a C++ program to implement the Stack ADT using the Single Linked List.

PART B: Non-Linear Data Structure

10. Write a C++ program to perform the following operations.
    a. Insert an element into a binary search tree.
    b. Delete an element from a binary search tree.
    c. Search for a key element in a binary search tree.
11. Write a C++ program to implement Bubble Sort.
12. Write a C++ program to implement Selection Sort.
13. Write a C++ program to implement Quick Sort.
14. Write a C++ program to implement Insertion Sort.
15. Write a C++ program for implementing the Merge Sort.
16. Write a C++ program for implementing the Heap Sort.
17. Write a C++ program that use recursive function to traverse the given binary tree in.
    a. Pre-order
    b. In order
    c. post-order
18. Write a C++ program to perform the following operations.
    a. Insertion into a B-tree.
    b. Deletion into a B-tree.
19. Write a C++ program to perform the following operations.
    a. Insertion into AVL tree.
    b. Deletion from a AVL tree.
20. Write a C++ program to implement Strassen's matrix multiplication.
21. Write a C++ program to implement Floyd's Warshall's Algorithm.
22. Write a C++ program to print all the nodes reachable from a given starting node in a diagram using BFS method.
23. Write a C++ program to check whether a given graph is connected or not using DFS method.
24. Write a C++ program to implement Brute Force String Matching algorithm.

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test of 20 marks shall be awarded.

Year	I	**Course Code:** 24MSCSC 2.5	**Credits**	04
Semester	II	**Course Title:** DATA COMMUNICATION AND NETWORK SECURITY	**Hours**	52
Course Pre-requisites if any	Knowledge of Basic computer networks			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***The objective of this course is to emphasizes the goal of*** o Understand the basics of networking principles, protocols, and architectures. o Learn about data transmission, routing, switching, and network protocols such as TCP/IP. o Explore security threats, vulnerabilities, and countermeasures in computer networks. o Study cryptographic algorithms, encryption techniques, and their applications in network security. o Gain skills in configuring, managing, and troubleshooting network devices and services to ensure network availability and reliability.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Gain a thorough understanding of data communication fundamentals, including protocols, standards, and network models, facilitating the design and implementation of network architectures. o Develop proficiency in analyzing data and signals, addressing transmission impairments, and optimizing data transmission rates for efficient communication. o Acquire knowledge of physical layer technologies and media, enabling effective selection and deployment of transmission media for different network environments. o Master network layer concepts such as logical addressing, IPv4, IPv6, and transport layer protocols like TCP and UDP, enhancing the ability to design and manage network communication. o Understand network security principles, including cryptography, authentication, access control, and intrusion detection, to ensure the integrity, confidentiality, and availability of network resources.

Unit No.	Course Content	Hours
UNIT I	**Introduction:** Data Communications, Networks, the internet, protocols and standards, network models– OSI model, TCP/IP protocol suite, addressing. Data and Signals: Periodic analog signals, digital signals, transmission impairment, data rate limits, performance.	10
UNIT II	**Physical Layer and Media: Analog transmission:** Digital-to-analog conversion, analog-to-analog conversion. Multiplexing, Transmission media – Guided media and unguided media. **Data Link Control:** Framing, flow, and error control.	12
UNIT III	**Network Layer:** Logical addressing – IPV4, IPV6, Address mapping–ARP, RARP, **Transport Layer:** Process to Process Delivery, User Datagram Protocol, Transmission Control Protocol, SCTP, Congestion Control.	10

UNIT IV	**Network Security:** Introduction, Types of Attacks, Fundamentals of Cryptography, Algorithms and Applications (Digital Signatures, Certificates and PKI (Public Key Infrastructure)). **Authentication and Access Control:** Authentication methods, Access control methods (DAC, MAC, RBAC).	10
UNIT V	**Email privacy:** Pretty Good Privacy (PGP) and S/MIME. IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management. Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS), Secure Electronic Transaction (SET). Intruders, Viruses, and related threats. Firewall Design principles, Trusted Systems. Intrusion Detection Systems.	10

Recommended Learning Resources

Textbooks:

1. Behrouza A Forouzan, Data Communications and Networking, McGraw-Hill.
2. Computer Networks - Andrew s. Tanenbaum, Pearson Education
3. Cryptography and network Security, Third edition, Stallings, PHI/ Pearson 2011
4. Principles of Information Security, Whitman, Thomson.2010
5. Network Security: The complete reference, Robert Bragg, Mark Rhodes, TMH 2010
6. Introduction to Cryptography, Buchmann, Springer 2012

Reference Books:

1. Data and Computer Communications, William Stallings, Pearson education
2. Data Communications, Computer Networks and Open Systems, fourth edition-Fred Halsall, Addison Wesley.
3. Network Security Essentials (Applications and Standards) by William Stallings Pearson Education, 5th Edition 2013.
4. Computer Networking a Top-Down Approach, K-Groas and Ross

Year	I	**Course Code:** 24MSCOE 2.6	**Credits**	04
Semester	II	**Course Title:** COMPUTER FUNDAMENTALS AND ITS APPLICATIONS	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***The objective of this course is to emphasizes the goal of*** o Understand the fundamentals of computers, including their evolution, types, and basic organization. o Gain knowledge of number systems, computer codes, Boolean algebra, and types of software and languages used in computing. o Learn about the anatomy and functioning of computers, including CPU, memory, input/output devices, and microcontrollers. o Explore operating system fundamentals, including functions, types, and basic commands in Unix. o Introduce database management systems (DBMS) and SQL, covering data types, SQL commands, and the role of DBMS in managing data efficiently.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Upon completion of this course, the student will be describing the components of a typical Computer and explain the characteristics of each of them. o Understand the working of Windows operating system and the services it provides. o Understand the importance of computers in business and society. o Describe various types of networks network standards and communication software.

Unit No.	Course Content	Hours
UNIT I	**Fundamentals of Computers:** Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software –System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.	10
UNIT II	**Introduction to computers:** Characteristics of computers, Classification of Digital Computer Systems: Microcomputers, Minicomputers, Mainframes, Super computers. Anatomy of Computer: Introduction, Functions & Components of a Computer, Central Processing Unit, Microprocessor, Storage units, Input and output Devices. How CPU and memory works. Program execution with illustrative examples. Introduction to microcontrollers.	12

UNIT III	**Operating System Fundamentals:** Operating Systems: Introduction, Functions of an operating System, Classification of Operating Systems, System programs, Application programs, Utilities, The Unix Operating System, Basic Unix commands, Microkernel Based Operating System, Booting.	10
UNIT IV	**Introduction to Database Management Systems:** Database, DBMS, Why Database –File system vs DBMS, Database applications, Database users, Introduction to SQL, Data types, Classification of SQL-DDL with constraints, DML, DCL, TCL	10
UNIT V	**Internet Basics:** Introduction, Features of Internet, Internet application, Services of Internet, Logical and physical addresses, Internet Service Providers, **Domain Name System. Web Basics:** Introduction to web, web browsers, http/https, URL, HTML5, CSS	10

Recommended Learning Resources

Textbooks:

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication, 2010.
2. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC, March 2014, ISBN: 9781466587793

Reference Books:

1. J. Glenn Brook shear," Computer Science: An Overview", Addision-Wesley, Twelfth Edition, June 2017.
2. R.G. Dromey, "How to solve it by Computer", Prentice-Hall International Series in computer science, C.A.R. HOARE Series Editor, PHI,ISBN: 0-13-433995-9.

Year	I	**Course Code:** 24MSCOE 2.6	Credits	04
Semester	II	**Course Title:** INTERNET CONCEPTS AND WEB DESIGN	Hours	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** o Understand foundational concepts of the Internet, including networks, protocols, and services. o Develop practical skills in creating web pages using HTML, including links, forms, and basic formatting. o Gain proficiency in CSS for styling web pages, including layout, fonts, and colors. o Introduce JavaScript for enhancing web interactivity with client-side scripting. o Acquire knowledge of network basics, including LANs, WANs, and network security principles.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Demonstrate a thorough understanding of Internet concepts and web protocols. o Build and style web pages using HTML, CSS, and JavaScript. o Utilize web development tools to create and manage websites. o Apply responsive design techniques to ensure websites are mobile-friendly. o Create interactive web applications with a focus on user experience and usability.

Unit No.	Course Content	Hours
UNIT I	**Concept of Internet:** A brief Introduction to the Internet, Computer Networks, Internet, URL (Uniform Resource Locator), Internet Service Provider, Intranet, Extranet, Virtual Private Network. Application of Internet: World Wide Web, Search Engines, News groups, Electronic Mail, Web Portal, Chat, Video Conferencing, FTP, Remote Login, E-Commerce, E-Learning, E-Governance, E-Banking.	10
UNIT II	**Internet Network:** Network definition, Common terminologies: LAN, WAN, Node, Host, Workstation, bandwidth, Interoperability, Network administrator, network security, Network Components: Severs, Clients, Communication Media, **Types of networks:** Peer to Peer, Clients Server, **Addressing in Internet:** DNS, Domain Name and their organization, understanding the Internet Protocol Address. Network topologies: Bust, star and ring, Ethernet, FDDI, ATM and Intranet	12
UNIT III	**Basics of HTML:** What is Internet Language? Understanding HTML, Create a Web page, Linking to other Web Pages, Publishing HTML Pages, Text Alignment and Lists, Text Formatting Fonts Control, Email Links and link within a Page, Creating a Table, Creating HTML Forms, Creating Web Page Graphics, Putting Graphics on a Web Page, Custom Backgrounds and Colours, Creating Animated Graphics.	10

UNIT IV	**Concept of CSS,** Creating Style Sheet ,CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties),Navigation Bar, CSS Colour, Creating page Layout and Site Designs.	10
UNIT V	**Introduction to JavaScript:** Writing First Java Script, External JavaScript, Variables: Rules for variable names, Declaring the variable, assign a value to a variable, Scope of variable, Using Operators, Control Statements, JavaScript loops, **JavaScript Functions:** Defining a Function, returning value from function, User define function.	10

Recommended Learning Resources

Textbooks:

1. Greenlaw R and Hepp E "Fundamentals of Internet and www" 2nd EL, Tata McGrawHill, 2007.
2. Ivan Bayross, "HTML, DHTML, JavaScript, Perl CGI", 3rd Edition, BPB Publications. D. Comer, "The Internet Book", Pearson Education, 2009.
3. Internet and Web Design Based on DOEACC III Revised syllabus 'O' Level Mac Millan India Ltd.
4. Teach Yourself HTML 4 in 24 Hours By Dick Oliver (Tech media) 4th edition
5. The Complete Reference JavaScript, By Thomas Powell & Fritz Schneider 2nd Edition.
6. Introduction to Internet and HTML Scripting By Bhaumik Shroff, Books India Publication.

Reference Books:

1. HTML and CSS By Dick Oliver and Michael Morrison (Pearson Education) 7th edition
2. HTML, DHTML, JavaScript, Perl CGI By Ivan Bayross(BPB) 3rd Edition
3. CSS By Kynn Bartlett (Pearson Education)2nd Edition
4. Introduction to Internet & HTML Scripting By Bhaumik Shroff Books India Publication 3rd Edition.

Syllabus of III Semester M.Sc. Programme

Choice Based Credit System (CBCS)

(According to new regulations w.e.f.2020-21)

Semester-III	III SEMSTER M.Sc. w.e.f. 2025-2026								
	Course	Subject Name	Teaching Hrs./ week	Practical Hrs./ week	Examination				Credits
					Duration (Hrs.)	Marks			
						Theory/ Practical	IA	Total	
Core Subject	24MSCCS 3.1	Web Programming	4	--	3	80	20	100	4
	24MSCCS 3.2	Python Programming	4	--	3	80	20	100	4
	24MSCCSPL 3.3	Web Programming Lab	--	4	3	80	20	100	4
	24MSCCSPL 3.4	Python Programming-Lab	--	4	3	80	20	100	4
Soft Core / Specialization/ Optional	24MSCSC 3.5	Software Engineering and Design	4	--	3	80	20	100	4
Note: Students has to Choose any one Soft Core / Specialization/ Optional									
Core Elective	24MSCCE 3.6	a. Internet of Things b. Operating System	4	--	3	80	20	100	4
Note: Students has to Choose any one Core Elective									
	Total		20	8				700	24

CS: Core Subject SC: Soft Core PL: Practical CE: Core

Year	II	**Course Code:** 24MSCCS 3.1	**Credits**	04
Semester	III	**Course Title:** DATABASE MANAGEMENT SYSTEMS	**Hours**	52

Course Pre-requisites if any	N/A	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***Objective of this course is to emphasizes the goal of*** ○ Understand the fundamental concepts of databases, including data models (relational, NoSQL), schemas, and normalization. ○ Develop proficiency in SQL (Structured Query Language) for database querying, manipulation, and data definition. ○ Learn to design databases, including schema design, entity-relationship modeling, and database normalization. ○ Explore techniques for ensuring data integrity, implementing access controls, and securing databases. ○ Understand transaction processing, concurrency issues, and techniques for ensuring ACID properties in database systems.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Understand the fundamental concepts of databases, including the characteristics and advantages of the database approach, and the three-schema architecture. ○ Gain proficiency in the relational model, including concepts, constraints, relational database schemas, and relational algebra operations. ○ Master relational database design principles, including normalization techniques and transaction processing concepts. ○ Acquire practical skills in SQL for data definition, manipulation, and complex querying, including nested and correlated queries. ○ Learn database recovery techniques, database security, and authorization mechanisms, ensuring data integrity and confidentiality in database systems.

Unit No.	Course Content	Hours
UNIT I	**Introduction:** Introduction to database, Characteristics of Database approach, Advantages of using DBMS approach, Three-schema architecture and data independence, **Entity-Relationship Model:** Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationships, Relationship types; Roles and Structural Constraints; Weak Entity Types. ER-Relational Mapping Rules.	10
UNIT II	**Relational Model:** Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION.	10

UNIT III	**Relational Database Design:** Anomalies in a database, functional dependency, and normal forms, lossless join, and dependency, BCNF, normalization through synthesis, higher order normal forms. **Transaction Processing Concepts:** Introduction to Transaction processing, Transaction and System concepts, Desirable properties of Transactions and issues with concurrent transactions.	10
UNIT IV	**SQL:** SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL Queries; Nested and Correlated Queries, IN, ALL, EXIST operators. Insert, Delete and Update statements in SQL.	10
UNIT V	**Database recovery techniques** based on deferred up data and immediate updating, shadow pages, ARIES recovery algorithm, database security and authorization, security issue access control based on granting/revoking of privileges, introduction of statistical database security.	12

Recommended Learning Resources

Textbooks:

1. "Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke (Publisher: McGraw-Hill Education; Year: 2002)
2. "Database Systems: The Complete Book" by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom (Publisher: Pearson; Year: 2009)
3. "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe (Publisher: Pearson; Year: 2015)
4. "SQL Performance Explained" by Markus Winand (Publisher: Markus Winand; Year: 2012)

Reference Books:

1. "An Introduction to Database Systems" by Bipin C. Desai (Publisher: Galgotia Publications; Year: 1987)
2. "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe (Publisher: Addison-Wesley; Year:1989)
3. "SQL for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL" by John L. Viescas and Michael J. Hernandez (Publisher: Addison-Wesley Professional; Year: 1996).

Year	II	**Course Code:** 24MSCCS 3.2	**Credits**	04
Semester	III	**Course Title:** PYTHON PROGRAMMING	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** o Master the fundamentals of Python syntax, data types, variables, and basic operations. o Understand control structures such as loops (for, while), conditional statements (if, else), and exception handling in Python. o Learn to define and use functions, organize code into modules, and import modules for code reuse. o Explore built-in data structures in Python such as lists, tuples, dictionaries, and sets, and understand their usage and operations.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Utilize core Python syntax and semantics: Confidently write Python programs using fundamental data types, control flow (if-else, loops), and functions. o Implement object-oriented programming concepts: Apply basic principles of object-oriented programming (OOP) in Python, including classes, objects, and methods. o Work with data structures and algorithms: Utilize fundamental data structures (lists, dictionaries) and basic algorithms to solve computational problems effectively. o Handle input/output and exceptions: Implement input/output functionalities (reading from files, user interaction) and basic exception handling for robust programs. o Analyze and apply problem-solving strategies: Break down problems into smaller steps, design algorithms using Python constructs, and analyze program efficiency.

Unit No.	Course Content	Hours
UNIT I	Installing Python, Simple program using Python, Expressions and Values, Variables and Computer Memory, error detection, Multiple line statements, Designing and using functions, functions provided by Python, omitting return statement. **Working with Text:** Creating Strings of Characters, Using Special Characters in Strings, creating a Multiline String, Printing Information, Getting Information from the Keyboard.	10
UNIT II	A Boolean Type, Choosing Statements to Execute, Nested If Statements, Remembering the Results of a Boolean Expression Evaluation, A Modular Approach to Program Organization, Importing Modules, Defining Your Own Modules, Testing Code Semi Automatically Grouping Functions Using Methods: Modules, Classes, and Methods, Calling Methods the Object-Oriented Way, Exploring String Methods, Underscores.	10

UNIT III	**Storing Collections of Data Using Lists:** Storing and Accessing Data in Lists, modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. **Repeating Code Using Loops:** Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops Using Break and Continue Reading and Writing.	10
UNIT IV	**Files:** Kinds of files, opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, and Writing Algorithms That Use the File-Reading Techniques, Multiline Records. Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, inverting a Dictionary, Using the In Operator on Tuples, Sets, and Dictionaries, Comparing Collections.	10
UNIT V	**Introduction to Pandas and Data Structures:** Overview of Pandas and its importance in data analysis. Installing Pandas. Introduction to Series and DataFrame - Pandas' primary data structures. Creating Series and DataFrame objects. Basic operations and attributes of Series and DataFrame (e.g., indexing, slicing, shape, size). **Data Manipulation with Pandas:** Loading data from various sources (CSV, Excel, SQL, etc.). Inspecting and cleaning data (handling missing values, duplicates). Data selection and filtering. Modifying data (adding, removing, and updating rows/columns). Sorting and ranking data. **Data Analysis and Visualization:** Descriptive statistics (mean, median, mode, standard deviation, etc.). Grouping and aggregation operations. Combining and merging, Basic data visualization with Pandas (using plot () function). Exporting data to different formats. **Advanced Topics and Practice:** Time series data analysis with Pandas. Working with categorical data. Handling large datasets efficiently (memory optimization). Reshaping and pivoting data. Additional resources for further learning and practice.	12

Recommended Learning Resources

Textbooks:

1. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
2. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey, Jeffrey Elkner, 2015.

Reference Books:

1. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr
2. Exploring Python, Timothy A. Budd, McGraw Hill Education
3. Python for Informatics: Exploring Information, Charles Severance.
4. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication.

Year	II	**Course Code:** 24MSCCSPL 3.3	**Credits**	03
Semester	IV	**Course Title:** DATABASE MANAGEMENT SYSTEMS LAB	**Hours**	04 Hr/lab
Course Pre-requisites if any	N/A			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***Objective of this course is to emphasizes the goal of*** o Develop practical skills in writing SQL queries for database manipulation and retrieval. o Apply concepts learned in theory to design and implement databases using a DBMS (Database Management System). o Learn techniques to optimize SQL queries for performance using indexes, query rewriting, and execution plans. o Practice designing database schemas, entity-relationship diagrams (ERDs), and normalization techniques. o Apply learned concepts to real-world scenarios through hands-on projects and assignments.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Design and manage relational databases using a DBMS. o Write and execute SQL queries for data manipulation and retrieval. o Implement data integrity, security, and access control measures. o Utilize database administration tools for backups, users, and performance. o Gain practical experience working with database systems.

PART A:

1. **School Management System:**
   Develop a database for a school with tables for students, classes, teachers, and grades. Students should write SQL queries to:
   - List students enrolled in a specific class.
   - Calculate average grades for a class or individual student.
   - Search for students by name or grade level.
   - Update student information and class enrolments.
2. **Library Database with JOINs:**
   Create a database for a library with tables for books, authors, genres, and borrowed books. Students should learn about JOIN operations in SQL to:
   - Retrieve book information, including author and genre, based on a book title.
   - Find books available for borrowing by checking borrowed books table.
   - Search for books by author or genre using JOINS.
   - Generate reports on the most popular books based on borrowing history.
3. **Movie Database with Filters:**
   Design a database for a movie database with tables for movies, actors, directors, and genres. Students should write SQL queries using WHERE clause and filtering techniques to:
   - List movies released in a specific year.

- Find movies with a particular rating (e.g., above 4 stars).
- Search for movies by keywords in the title or description.
- Implement basic filtering options for genre and director.

4. **Restaurant Menu Management:**
   Design a database for a restaurant with tables for menu items, categories, ingredients, and prices. Students should write SQL queries to:
   - List menu items by category.
   - Search for dishes by keywords in the name or description.
   - Update ingredient stock levels based on recipe requirements.

Generate reports on the most popular menu items and revenue generated.

PART B:

5. **University Course Enrolment:**
   Develop a database for a university course registration system. Students should write SQL queries to:
   - List available courses for a semester.
   - Allow students to register for courses (assuming no waitlists or restrictions).
   - Generate basic reports on course enrolment numbers.
   - Integrate with a simple interface for course listings and registration.
6. **Hospital Patient Management:**
   Design a database for a hospital with tables for patients and basic medical information. Students should write SQL queries to:
   - Search for patients by name or ID.
   - Update patient information (e.g., contact details).
   - Generate basic reports on patient demographics.
7. **Travel Booking System (Limited Functionality):**
   Develop a database for a travel booking system with tables for destinations and basic information like attractions or hotels. Students should write SQL queries to:
   - Search for destinations based on keywords or criteria (e.g., location, type).
   - Display basic information about available destinations.
   - (Optional) Integrate with a basic interface for destination search and display.
8. **Blog Management System (Core Functionality):**
   Design a database for a blog with tables for blog posts, categories, and authors. Students should write SQL queries to:
   - List published blog posts by category.
   - Search for blog posts by keywords in the title or content.
   - Allow basic user accounts and post creation (without extensive editing/approval functionalities).
   - Integrate with a basic interface for displaying blog posts and search functionality.

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test of 20 marks shall be awarded.

Year	II	**Course Code:** 24MSCCSPL 3.4	**Credits**	03
Semester	III	**Course Title:** PYTHON Programming LAB	**Hours**	04 Hr/lab

Course Pre-requisites if any	N/A	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** o Gain proficiency in writing Python code, understanding syntax, data types, variables, and basic operations. o Practice using control structures such as loops (for, while), conditional statements (if, else), and exception handling to control program flow. o Develop skills in defining and using functions, organizing code into modules, and importing modules for code reuse and organization. o Implement and manipulate built-in data structures like lists, tuples, dictionaries, and sets, and apply algorithms for sorting, searching, and data manipulation.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Get the Proficiency in Core Python Programming o Problem Solving and Data Manipulation o Competent in File Handling and Exception Management o Skilled in Data Analysis and Visualization o Experienced in Practical Applications and Advanced Techniques

Practice Lab programs:

1. Implement Python Script to generate first N natural numbers.
2. By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms.
3. Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.
4. Write a function cumulative product to compute cumulative product of a list of numbers.
5. Write a function reverse to reverse a list. Without using the reverse function.
6. Define a function which generates Fibonacci series up to n numbers using RECURSION.
7. With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.
8. Write a program to count the numbers of characters in the string and store them in a dictionary data structure.
9. Remove spaces from a string using recursion.
10. Write a program for Set Operations: Union, Intersection, and Difference
11. Write a program to compute File Content Analysis: Line, Word, and Character Counts
12. Write a program to compute Error Handling in User Input and Division Operations

13. Write a program to compute CSV Data Analysis using Pandas
14. Write a program to perform the Data Visualization with Matplotlib: Line and Bar Charts
15. Write a program for Email Validation using Regular Expressions.
16. Write a program to compute Function Logging and Prime Number Generation with Decorators and Generators

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test, 20 marks shall be awarded.

Year	II	**Course Code:** 24MSCSC 3.5	**Credits**	04
Semester	III	**Course Title:** SOFTWARE ENGINEERING AND DESIGN	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		
Course Objective	***The objective of this course is to emphasizes the goal of*** o Understand the phases of the software development lifecycle (SDLC) and different methodologies like Agile, Waterfall, and DevOps. o Learn techniques for eliciting, analyzing, and managing software requirements to ensure alignment with stakeholder needs. o Apply principles of software design such as modularity, abstraction, encapsulation, and SOLID principles. o Gain skills in software testing techniques, test automation, and quality assurance processes to ensure software reliability and quality. Understand project management techniques specific to software engineering, including scheduling, budgeting, and resource management for successful project delivery.			
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Apply software development life cycle (SDLC) methodologies to design, develop, and test software systems. o Utilize fundamental software engineering principles for problem-solving, code design, and maintainability. o Implement core user interface (UI) and user experience (UX) design concepts for creating intuitive and user-friendly software. o Effectively communicate software design decisions and technical documentation using appropriate tools and practices. o Analyze and apply software engineering best practices for quality assurance, version control, and project management.			

Unit No.	Course Content	Hours
UNIT I	**Introduction:** Professional software development, Software engineering ethics, Case studies. Software Processes: Software Process models, Process activities: Software specification, Software design and implementation, Software reliability and availability, software management activities- Managing people, software cost estimation, process, improvement, configuration management.	10
UNIT II	**Requirements Engineering:** Functional and non-functional requirements, introduction to Requirements specification. Agile Software Development: Agile methods- Plan driven and Agile Development, Introduction to Extreme Programming.	12
UNIT III	**Design and Implementation:** Object-oriented design using UML: System Context and Interaction, Architectural design, Object Class identification, design Models, Interface Specification, Design Patterns, Implementation issues, Open-Source development.	10

UNIT IV	**Verification, Validation and Management:** Software inspections, static analysis, verification and formal methods, software testing- Unit Testing, Choosing Unit Test Cases, Component Testing, System Testing, Test Driven Development, critical systems validation.	10
UNIT V	**Quality Management:** Introduction, Software quality, Software standards: The ISO 9001 standard framework, Reviews, and inspection. Configuration management: Introduction to Change management, Version management, System building, Release management.	10

Recommended Learning Resources

Textbooks:

1. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin, Publisher: Prentice Hall, Year: 2008
2. "Code Complete: A Practical Handbook of Software Construction" by Steve McConnell, Publisher: Microsoft Press, Year: 2004
3. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Publisher: Addison-Wesley Professional, Year: 1994

Reference Books:

1. “Software Engineering-A Practitioners approach”, by Roger. S. Pressman: 8th Edition and above, Tata McGraw Hill
2. “Software Testing Craftsman’s Approach”, by Paul C. Jorgensen, 4th Edition CRC Press, Taylor Francis Group
3. “Fundamentals of Software Engineering” by Rajib Mall, 4th Edition onwards PHI Learning Pvt. Ltd.
4. “An Integrated Approach to Software Engineering” by Pankaj Jalote, Wiley India, 2009 onwards Resources.

Year	II	**Course Code:** 24MSCCE 3.6	**Credits**	04
Semester	III	**Course Title:** INTERNET OF THINGS (IOT)	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***Objective of this course is to emphasizes the goal of*** o Understand the basic concepts, architecture, and components of the Internet of Things. o Learn about IoT devices, sensors, actuators, and their integration with cloud platforms. o Explore communication protocols used in IoT, such as MQTT, CoAP, and HTTP, for device-to-device and device-to-cloud communication. o Study security challenges and privacy issues in IoT deployments, and techniques for securing IoT devices and data. o Analyze real-world applications of IoT in various domains such as smart cities, healthcare, agriculture, and industrial automation.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Understand the fundamental principles of IoT, including its components (sensors, actuators, and communication), architecture, and applications in various domains. o Utilize sensors, actuators, and microcontrollers to design and build simple IoT prototypes for data collection and interaction. o Understand and implement common communication protocols (e.g., Wi-Fi, Bluetooth) used for data transmission in IoT networks. o Utilize data analysis tools to process, visualize, and extract insights from sensor data collected by IoT devices. o Identify potential security and privacy risks in IoT systems and propose mitigation strategies.

Unit No.	Course Content	Hours
UNIT I	**Fundamentals of IoT:** Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.	12
UNIT II	**Sensors Networks:** Definition, Types of Sensors, Types of Actuators, Examples and Working, **IoT Development Boards:** Arduino IDE and Board Types, Raspberry Pi Development Kit, RFID Principles and components, Wireless Sensor Networks: History and Context, The node, Connecting nodes, Networking Nodes, WSN and IoT.	10
UNIT III	**Communicating smart objects:** Communication criteria, IoT access technologies- IEEE 802.15.4, IEEE 802.15.4e, IEEE 802.11ah, IEEE 1901.2a, NB-IoT. **IoT Network Layer:** IP as IoT network layer, 6LoWPAN, 6Lo, 6TiSCH, RPL. IoT Application Layer: IoT application transport methods, CoAP, MQTT. IP Based Protocols for IoT IPv6, 6LowPAN, RPL, REST, AMPQ, CoAP, MQTT. Edge connectivity and protocols.	10

UNIT IV	**Data and Analytics for IoT:** IoT Middleware, Data analytics for IoT, Big Data analytics tools and technology. Introduction to Hadoop. Introduction to data Analytics, Types of Data analytics, Local Analytics, Cloud analytics and applications.	10
UNIT V	**IoT application case study:** Smart City, Smart Grid, Smart Transportation, Smart Manufacturing, Smart Healthcare, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.	10

Recommended Learning Resources

Textbooks:

1. Hakima Chaouchi, — "The Internet of Things Connecting Objects to the Web" ISBN :978-1-84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, — "The Internet of Things: Key Applications and Protocols", WileyPublications
3. Vijay Madisetti and ArshdeepBahga, — "Internet of Things (A Hands-on-Approach)",1st Edition, VPT, 2014.
4. J. Biron and J. Follett, "Foundational Elements of an IoT Solution", O'Reilly Media, 2016.
5. Keysight Technologies, "The Internet of Things: Enabling Technologies and Solutionsfor Design and Test", Application Note, 2016.

Reference Books:

1. Daniel Minoli, — "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications
2. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies,Platforms, and Use Cases", CRC Press
3. https://onlinecourses.nptel.ac.in/noc17_cs22/course
4. http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html

Year	II	**Course Code:** 24MSCCE 3.6	**Credits**	04
Semester	III	**Course Title:** OPERATING SYSTEM	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	*Objective of this course is to emphasizes the goal of* o Grasp foundational concepts like processes, threads, memory management, and file systems. o Familiarize with system call mechanisms and API interactions for OS functionality. o Learn techniques to manage concurrent processes and synchronize access to shared resources. o Explore memory allocation strategies, virtual memory, and memory protection mechanisms. o Study file system structures, file operations, and input/output device management.
Course Outcomes	*After completing this course satisfactorily, a student will be able to:* o Understand the fundamental concepts of operating systems, including system structure, process management, and memory management. o Develop proficiency in process scheduling algorithms, synchronization techniques, and deadlock handling strategies. o Gain practical skills in memory management strategies, file system operations, and Unix commands for efficient system operation. o Learn shell programming essentials, including scripting, conditional statements, loops, and debugging techniques, through hands-on laboratory sessions. o Apply theoretical knowledge to implement programs that supplement course concepts, enhancing understanding and practical skills in operating system concepts.

Unit No.	Course Content	Hours
UNIT I	**Introduction to Operating Systems:** System Structure What operating systems do; Computer System Organization; Computer System Architecture; Operating System Operations; Computing Environments; Operating System Services; System Calls; Types of System Calls; System Programs; Operating System Structure; Virtual Machines; System boot. **Overview of Process Concept**: Process Scheduling; Operations on Processes; Inter – Process Communication; Multi – Threaded Programming: Overview: Multithreading Models.	12
UNIT II	**Process Management Process Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple Processor Scheduling. Process Synchronization, The Critical Section Problem; Peterson's Solution; Semaphores; Classical Problems of Synchronization.	10

UNIT III	**Deadlocks:** System model; Deadlock Characterization, Methods for handling deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock Detection and Recovery from deadlock. **Memory Management Strategies:** Background, Swapping; Contiguous Memory Allocation; Paging; Segmentation; Virtual Memory Management; Background; Demand Paging; Page Replacement; Allocation of Frames; Thrashing.	10
UNIT IV	**The File System:** The File, What's in a File name? The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, The Unix File System. The vi Editor: vi Basics, Input Mode, ex Mode and Command Mode. Basic File Attributes: ls options, File Ownership, File Permissions, chmod, Directory Permissions. **Changing the File Ownership More File Attributes:** File Systems and Inodes, Hard Links, Symbolic Links, The Directory, umask, Modification and Access Times, find. **The Shell**: The Shell's Interpretive Cycle, Shell Offerings, Pattern Matching- The Wildcards, Escaping and Quoting, Redirection: The Three Standard Files, Two Special Files: /dev/null and /dev/tty, pipes, tee: Creating a Tee, Command Substitution.	10
UNIT V	**The Process:** Process Basics, PS: Process Status, System Processes, Mechanism of Process Creation, Internal and External Commands, Running Jobs in Background, Killing Processes with Signals, Job Control, at and batch, cron. **Essential Shell Programming:** Shell Variables, Environment Variables, Shell Scripts, read, Using Command Line Arguments, exit and exit status of command, 16 The Logical Operators, The if Conditional, using test and [] to Evaluate Expression, The case Conditional, expr, while: looping, for: looping with a list, set and shift, trap, Debugging Shell Scripts with set – x Laboratory Students shall implement programs which supplement the theory concepts.	10

Recommended Learning Resources

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating Systems Principles, 8th Edition, Wiley – India.

Reference Books:

1. D M Dhamdhere: Operating Systems – A Concept Based Approach, 2nd Edition, Tata McGraw – Hill, 2002.
2. P C P Bhatt: Operating Systems, 2ndEdition, PHI, 2006. Harvey M Deital: Operating Systems, 3rd Edition, Addison Wesley, 1990.

Syllabus of IV Semester MSC programme,
Choice Based Credit System (CBCS)
(According to new regulations w.e.f. 2020-21)

Semester-IV	IV SEMSTER MSC w.e.f. 2025-2026								
	Course	Subject Name	Teaching Hrs./ week	Practical Hrs./ week	Examination				Credits
					Duration (Hrs.)	Marks			
						Theory/ Practical	IA	Total	
Core Subject & Electives	24MSCCS 4.1	Digital Image Processing	4hrs	--	3	80	20	100	4
	24MSCCS 4.2	Artificial Intelligence and Machine Learning	4hrs	--	3	80	20	100	4
	24MSCCSPL 4.3	Digital Image Processing LAB	--	4hrs	3	80	20	100	4
	24MSCCE 4.4	a. Software Testing b. Cloud Computing c. Big Data Analytics	4hrs	--	3	80	20	100	4
Soft Core/ Specialization / Optional	24MSCSC 4.5	Data Mining Techniques	4hrs	--	3	80	20	100	4
Project Work	24MSCPJ 4.6	Project Work	-	4hrs	3	80	20	100	4
	Total		16	8				600	24

CS: Core Subject SC: Soft Core PL: Practical Lab OE: Open Elective PJ: Project

Year	II	**Course Code:** 24MCACS 4.1	**Credits**	04
Semester	IV	**Course Title:** DIGITAL IMAGE PROCESSING	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** ○ Understand the basic principles and characteristics of digital images, including pixels, color models, and resolution. ○ Learn techniques for improving the quality of digital images through operations such as contrast adjustment, noise reduction, and sharpening. ○ Study methods for restoring degraded images by removing noise, blurring, or other distortions using filtering and restoration algorithms. ○ Explore methods for reducing the size of digital images while preserving image quality through compression techniques like JPEG, PNG, and HEVC. ○ Develop skills in extracting meaningful features from images and applying techniques for object detection, recognition, and classification.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Understand the fundamental concepts of digital images, their representation using pixels, and various image formats. ○ Utilize common image processing techniques for tasks like enhancement, filtering, and restoration to improve image quality. ○ Implement algorithms to extract meaningful features from images, useful for applications like object detection or image analysis. ○ Understand various image compression techniques to reduce file size while maintaining image quality. ○ Apply acquired knowledge to develop simple image processing applications for specific functionalities.

Unit No.	Course Content	Hours
UNIT I	**Digital Image Fundamentals:** The origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamentals Steps in Image Processing, Elements of Digital Image Processing Systems, Image Sampling and Quantization, Basic relationships between pixels..	10
UNIT II	**Image Enhancement in the Spatial Domain**: Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Basics of Spatial Filters, Smoothening and Sharpening Spatial Filters, Combining Spatial Enhancement Methods.	12
UNIT III	**Image Enhancement in the Frequency Domain**: Introduction to Fourier Transform and the frequency Domain, Smoothing and Sharpening Frequency Domain Filters, Homomorphic Filtering. Colour Image Processing: Colour models, pseudocolor image processing, colour transformations, smoothening and sharpening.	10

UNIT IV	**Image Restoration:** A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.	10
UNIT V	**Image Segmentation:** Detection of Discontinuities, Edge linking and boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation. Morphological Processing: Some basic Morphological operations.	10

Recommended Learning Resources

Textbooks:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 3/e, Pearson Education.
2. Anil K. Jain, Fundamentals of Digital Image Processing', Pearson
3. Kenneth R. Castleman, Digital Image Processing, Pearson.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, Digital Image Processing using MATLAB, Pearson Education, Inc.

Reference Books:

1. William K. Pratt, Digital Image Processing, John Wiley, New York,
2. Milan Sonka et al, Image Processing, Analysis and Machine Vision, Brookes/Cole, Vikas Publishing House.

Year	II	**Course Code:** 24MSCCS 4.2	**Credits**	04
Semester	IV	**Course Title:** Artificial Intelligence and Machine Learning	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***objective of this course is to emphasizes the goal of*** o Understand the basic concepts, goals, and scope of artificial intelligence and machine learning. o Learn supervised, unsupervised, and reinforcement learning algorithms for pattern recognition, prediction, and decision-making. o Explore techniques for data cleaning, pre-processing, and feature extraction to prepare data for machine learning models. o Gain skills in evaluating machine learning models using performance metrics and optimizing models for accuracy, speed, and scalability. o Study real-world applications of artificial intelligence and machine learning in fields such as healthcare, finance, autonomous systems, and natural language processing.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Apply core AI concepts (search, reasoning) to problem-solving and intelligent agent design. o Implement search algorithms (BFS, DFS, A*) and analyze their effectiveness. o Utilize machine learning techniques (classifiers, k-means clustering) for data analysis. o Understand the fundamentals of supervised and unsupervised learning. o Explain key concepts in AI like problem formulation, reasoning methods, and machine learning applications.

Unit No.	Course Content	Hours
UNIT I	**Artificial Intelligence:** Application of AI, AI Problems, Problem Formulation, Intelligent Agents, Types of Agents, Agent Environments, PEAS representation for an Agent, Architecture of Intelligent agents. Reasoning and Logic, Prepositional logic, first order logic, Using First-order logic, Inference in First-order logic, forward and Backward Chaining.	12
UNIT II	**Search Strategies**: Solving problems by searching, Search- Issues in The Design of Search Programs, Un-Informed Search- BFS, DFS; Heuristic Search Techniques: Generate-And Test, Hill Climbing, Best-First Search, A* Algorithm, Alpha beta search algorithm, Problem Reduction, AO*Algorithm, Constraint Satisfaction, Means-Ends Analysis.	10
UNIT III	**Artificial Neural Networks:** Introduction, Activation Function, Optimization algorithm- Gradient decent, Networks Perceptron's, Adeline, Multilayer Perceptron's, Back propagation Algorithms Training Procedures, Tuning the Network Size.	10

UNIT IV	**Introduction to ML:** Machine Learning basics, Applications of ML, Data Mining Vs Machine Learning vs Big Data Analytics. Supervised Learning-Naïve Base Classifier, Classifying with k-Nearest Neighbor classifier, Decision Tree classifier, Naive Bayes classifier. Unsupervised Learning - Grouping unlabeled items using k-means clustering, Association analysis with the Apriori algorithm Introduction to reinforcement learning.	10
UNIT V	**Forecasting and Learning Theory:** Non-linear regression, Logistic regression, Random Forest, Bayesian Belief networks, Bias/variance tradeoff, Tuning Model Complexity, **Model Selection Dilemma Clustering:** Expectation-Maximization Algorithm, Hierarchical Clustering, Supervised Learning after Clustering, Choosing the number of clusters, Learning using ANN.	10

Recommended Learning Resources

Textbooks:

1. Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, Pearson, 2021
2. "Artificial Intelligence: Foundations of Computational Agents" by David L. Poole and Alan K. Mackworth, Cambridge University Press, 2017
3. "Artificial Neural Networks: Methods and Applications" by David Kriesel, CreateSpace Independent Publishing Platform, 2007
4. "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy, MIT Press, 2012
5. "Machine Learning Yearning" by Andrew Ng, self-published, 2018

Reference Books:

1. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving" by George F. Luger, Pearson, 2008
2. "Artificial Intelligence: Foundations of Computational Agents" by David L. Poole and Alan K. Mackworth, Cambridge University Press, 2017
3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop, Springer, 2006.

<table>
<tr><td><b>Year</b></td><td>II</td><td colspan="2"><b>Course Code:</b> 24MSCCSPL 4.3</td><td><b>Credits</b></td><td>03</td></tr>
<tr><td><b>Semester</b></td><td>IV</td><td colspan="2"><b>Course Title:</b> DIGITAL IMAGE PROCESSING LAB</td><td><b>Hours</b></td><td>04 Hr/lab</td></tr>
<tr><td colspan="2">Course Pre-requisites if any</td><td colspan="4">N/A</td></tr>
<tr><td colspan="2">Formative Assessment Marks: 20</td><td>Summative Assessment Marks: 80</td><td colspan="3">Duration of ESA: 03hrs.</td></tr>
<tr><td><b>Course Objective</b></td><td colspan="5"><i>The objective of this course is to emphasizes the goal of</i>
o Gain practical experience in applying basic image processing operations such as image enhancement, transformation, and filtering using MATLAB or Python.
o Implement algorithms to restore degraded images and reduce noise using techniques like spatial and frequency domain filtering.
o Practice methods for partitioning images into meaningful regions or objects using segmentation algorithms such as thresholding, region growing, and watershed.
o Develop skills in extracting features from images and applying techniques like edge detection, corner detection, and template matching for object recognition.
o Work on projects that integrate various image processing techniques to solve real-world problems or applications, fostering hands-on experience and practical skills.</td></tr>
<tr><td><b>Course Outcomes</b></td><td colspan="5"><i>After completing this course satisfactorily, a student will be able to:</i>
o Implement basic image processing algorithms: Apply techniques like filtering, histogram manipulation, and image enhancement using Python libraries or custom code.
o Analyze and visualize image data: Utilize tools to explore and visualize image properties (histograms, frequency domain) for better understanding.
o Perform image feature extraction: Implement image processing algorithms to extract features like edges, shapes, or textures from images.
o Develop simple DIP applications: Design and code basic image processing applications for specific tasks (e.g., noise reduction, image segmentation).
o Experiment and interpret results: Conduct experiments using DIP tools and libraries, analyze results, and draw conclusions from image processing operations</td></tr>
</table>

Programs

1. Perform all Arithmetic and Logical operations on any Natural image(s) in MATLAB/Python
2. Perform the image sampling and quantization in MATLAB/Python and notice the bits variations in a given input image.
3. Perform the Basic relationships of pixels along with the distance measures in MATLAB/Python
   a. Neighborhood
   b. Adjacency, Connectivity
   c. Paths, Regions and boundaries
   d. Distance Measures

4. Implement Interpolation methods used in Digital Image processing and implement these Methods in MATLAB/Python.
5. Perform the Smoothing and Sharpening on any medical image(s) in MATLAB/Python
6. Calculate Mean Square Error (MSE) and Peak signal-to-noise ratio (PSNR) of Noisy and
7. Noiseless images in MATLAB/Python and interpret your results for variations in MSE and PSNR.
8. Perform the Affine Transformations in MATLAB/Python
9. Perform the following operations in MATLAB/Python
   i. What will be the effect on histogram of an image, if we set the higher order bit plane to zero?
   ii. Perform the histogram processing on an image and visualize the effects.
10. Perform intensity slicing using Intensity Transformation Functions in MATLAB/Python

Instructions:

1. Certified Journal is mandatory for every student to appear the practical examination.
2. Based on the practical internal test, 20 marks shall be awarded.

Year	II	**Course Code:** 24MSCCE 4.4 (a)	**Credits**	04
Semester	IV	**Course Title:** SOFTWARE TESTING	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** o Grasp basic software testing concepts, including error and fault taxonomies and test case identification. o Understand and differentiate between various testing levels like unit, integration, and system testing. o Explore advanced testing techniques and challenges in interaction and object-oriented testing. o Learn and apply model-based testing and test-driven development (TDD) methodologies. o Utilize practical examples and tools for real-world software testing applications.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** o Gain a foundational understanding of software testing principles and methodologies, including test case development and error taxonomy. o Learn practical techniques for decision table-based testing and interaction testing, with applications in various software scenarios. o Understand levels of testing within different software development life cycles and their separation in integration and system testing. o Develop skills in object-oriented testing, focusing on integration and system testing using UML and specific frameworks. o Explore advanced testing methodologies such as model-based testing and all-pairs testing, along with practical applications like Test-Driven Development (TDD) in software engineering contexts.

Unit No.	Course Content	Hours
UNIT I	**Basics of Software Testing and Examples:** Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. **Examples:** Generalized pseudocode, The triangle problem, The NextDate function, The commission problem, The SATM problem. Decision Table-Based Testing: Decision tables.	10
UNIT II	**Levels of Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. **Integration Testing:** A closer look at the SATM system, Decomposition-based, call graph- based, Path-based integrations, Case study. **System Testing:** Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example.	12

UNIT III	**Interaction Testing:** Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing. **Issues in Object-Oriented Testing:** Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism,	10
UNIT IV	**Object-Oriented Integration Testing:** UML support for integration testing, MM-paths for object- oriented software, A framework for object-oriented dataflow integration testing. **GUI Testing:** The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program. **Object- Oriented System Testing:** Currency converter UML description	10
UNIT V	**Model-Based Testing:** Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. **Test-Driven Development:** Test-then-code cycles, Automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD. **A Closer Look at All Pairs Testing:** The all- pairs technique, A closer look at NIST study	10

Recommended Learning Resources

Textbooks:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2012.
2. Aditya P Mathur: Foundations of Software Testing, Pearson, 2008.
3. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons, 2011.

Reference Books:

1. SrinivasanDesikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 1st Edition, Pearson, 2012.
2. Brian Marrick: The Craft of Software Testing, 1st edition, Pearson, 2012

Year	II	**Course Code:** 24MSCCE 4.4 (b)	**Credits**	04
Semester	IV	**Course Title:** CLOUD COMPUTING	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** ○ Understand the basics of cloud computing models (IaaS, PaaS, SaaS), architecture, and deployment models. ○ Learn about popular cloud service providers (e.g., AWS, Azure, Google Cloud) and their offerings ○ Explore security challenges, best practices, and techniques specific to cloud environments. ○ Understand virtualization technologies (e.g., VMs) and containerization (e.g., Docker) used in cloud computing. ○ Gain hands-on experience in deploying applications, managing resources, and scaling infrastructure in the cloud
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Understand the fundamental characteristics and influences shaping cloud computing. ○ Gain knowledge of cloud delivery and deployment models, including SaaS, PaaS, and IaaS. ○ Learn essential cloud software security principles and secure development practices. ○ Identify and address various cloud computing risk issues and security challenges. ○ Comprehend the architectural considerations for creating a secure and trusted cloud environment.

Unit No.	Course Content	Hours
UNIT I	**Cloud Computing fundamentals:** Essential characteristics, Architectural Influences, Technological Influences, and Operational Influences.	10
UNIT II	**Cloud Computing Architecture:** Cloud Delivery models, The SPI Framework, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS), Cloud deployment models, Public Clouds, Community Clouds, Hybrid Clouds, Alternative Deployment models, Expected benefits.	12
UNIT III	**Cloud Computing Software Security fundamentals:** Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices, Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.	10

UNIT IV	**Cloud Computing Risk Issues:** The CIA Traid, Privacy and Compliance Risks, Threats to Infrastructure, Data and Access Control, Cloud Access Control Issues, Cloud Service Provider Risks. Cloud Computing Security challenges: Security Policy Implementation, Policy Types, and Computer Security Incident Response Team (CSIRT).	10
UNIT V	**Cloud Computing Security Architecture:** Architectural Considerations, General Issues, Trusted Cloud Computing, Secure Execution environments and Communications, Micro architectures, Identity Management and Access Control, Autonomic Security.	10

Recommended Learning Resources

Textbooks:

1. Anthony T. Vete, Toby J. Velte, Robert Elsenpeter, —Cloud Computing A Practical Approachǁ, McGraw-Hill, 2010.
2. Ronald L. Krutz, Russell Dean Vines, "Cloud Security A Comprehensive Guide to secure Cloud Computing" Wiley.

Reference Books:

1. John W. itinghouse james F.Ransome, "Cloud Computing Implementation, Management and Security", CRC Press.
2. Borko Furht. Armando Escalante, "Handbook of Cloud Computing", Springer
3. Charles Badcock, "Cloud Revolution", TMH

Year	II	**Course Code:** 24MSCCE 4.4 (c)	**Credits**	04
Semester	IV	**Course Title:** BIG DATA ANALYTICS	**Hours**	52
Course Pre-requisites if any	NA			
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.		

Course Objective	***The objective of this course is to emphasizes the goal of*** ○ Understand the characteristics, challenges, and opportunities presented by big data. ○ Learn about technologies and platforms used for big data processing and storage, such as Hadoop, Spark, and NoSQL databases. ○ Gain skills in processing and analyzing large-scale datasets using distributed computing frameworks. ○ Apply machine learning algorithms for predictive analytics and pattern recognition on big data. ○ Visualize and communicate insights from big data using appropriate tools and techniques for data visualization and dashboards
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Achieve proficiency in using big data technologies and analytical tools effectively. ○ Develop skills for efficient collection, storage, and management of large datasets. ○ Attain advanced capabilities in data processing and analysis using big data frameworks. ○ Demonstrate the ability to create meaningful and effective data visualizations. ○ Apply big data analytics techniques to solve complex real-world problems across various sectors.

Unit No.	Course Content	Hours
UNIT I	**Introduction to Big Data Processing:** Data Storage and Analysis - Characteristics of Big Data – Big Data Analytics - Typical Analytical Architecture – Requirement for new analytical architecture – Challenges in Big Data Analytics – Need of big data frameworks.	10
UNIT II	**HDFS (Hadoop Distributed File System):** The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures	12
UNIT III	**Map Reduce:** Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.	10
UNIT IV	**Introduction to Hadoop ecosystem technologies:** Serialization: AVRO, Co-ordination: Zookeeper, **Databases:** HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm.	10

UNIT V	**Data Analytics with R Machine Learning:** Introduction, Supervised Learning, Unsupervised Learning, Collaborative Filtering. Big Data Analytics with BigR.	10
Recommended Learning Resources		
Textbooks:		
1. Tom White “Hadoop: The Definitive Guide” Third Edit on, O’Reilly Media, 2012. 2. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.		
Reference Books:		
1. Mohammed Guller, Big Data Analytics with Spark, Apress,2015 2. Donald Miner, Adam Shook, “Map Reduce Design Pattern”, O’Reilly, 2012		

Year	II	**Course Code:** 24MSCSC 4.5	**Credits**	04
Semester	IV	**Course Title:** DATA MINING TECHNIQUES	**Hours**	52

Course Pre-requisites if any	NA	
Formative Assessment Marks: 20	Summative Assessment Marks: 80	Duration of ESA: 03hrs.

Course Objective	***The objective of this course is to emphasizes the goal of*** ○ Understand the basic concepts, goals, and processes of data mining. ○ Learn techniques for cleaning, transforming, and preprocessing data to prepare it for mining. ○ Explore various algorithms for supervised learning (classification, regression) and unsupervised learning (clustering, association rule mining). ○ Understand and apply evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of data mining models. ○ Study real-world applications of data mining techniques in areas such as marketing, healthcare, finance, and social media analysis.
Course Outcomes	***After completing this course satisfactorily, a student will be able to:*** ○ Understand basic principles and concepts of data mining. ○ Identify and apply appropriate data mining techniques to solve practical problems. ○ Evaluate the performance of data mining algorithms and interpret the results. ○ Gain practical experience in using data mining tools and software for analysis. ○ Develop the ability to communicate effectively about data mining concepts and results.

Unit No.	Course Content	Hours
UNIT I	**Introduction to Data Mining:** Related technologies - Machine Learning, DBMS, OLAP, Statistics, Data Mining Goals, Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods, Applications, Example: weather data.	10
UNIT II	**Data Warehouse and OLAP:** Data Warehouse and DBMS, Multidimensional data model, OLAP operations, Example: loan data set, **Data pre-processing:** Data cleaning, Data transformation, Data reduction, Discretization and generating concept hierarchies, Introduction to Weka Data Mining System, Example experiments with Weka - filters, discretization. Data mining knowledge representation: Visualization techniques, Experiments with Weka – visualization	12
UNIT III	**Attribute-oriented analysis:** Attribute generalization, Attribute relevance, Class comparison, Statistical measures, Experiments with Weka - using filters and statistics. **Data mining algorithms -Association rules:** Motivation and terminology, Example: mining weather data, Basic idea: item sets, Generating item sets and rules efficiently, Correlation analysis, Experiments with Weka - mining association rules.	10

UNIT IV	**Data mining algorithms- Classification:** Basic learning/mining tasks, inferring rudimentary rules: 1R algorithm, Decision trees, covering rules, Experiments with Weka - decision trees, rules. Data mining algorithms-Prediction: The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbor), linear models	10
UNIT V	**valuating what's been learned:** Basic issues, Training and testing, Estimating classifier accuracy (holdout, cross validation, leave-one-out), Combining multiple models (bagging, boosting, stacking), Minimum Description Length Principle (MLD), Experiments with Weka - **training and testing Clustering:** Basic issues in clustering, Cluster/2, Partitioning methods: k-means, expectation maximization (EM), **Hierarchical methods:** distance-based agglomerative and divisible clustering, Conceptual clustering: Cobweb, Experiments with Weka - k-means, EM, Cobweb.	10

Recommended Learning Resources

Textbooks:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann,
2. Jiaweihan, MichelineKamber,Jian Pei, Data mining concepts and techniques, 3/e, Elsevier

Reference Books:

1. Margaret H. Dunham, Data Mining-Introductory and Advanced Topics, Pearson Education
2. K.P.Soman, ShyamDiwakar, and V. Ajay, Insight into Data Mining: Theory and Practice, Prentice Hall of India,2006.

Course code: 24MSCPJ 4.6 PROJECT WORK

Project-IA(20 Marks)

Project / Technical Seminar:

A research committee, consisting of the Department Chairman, Guide/co-guide (if applicable), and a senior faculty member, will assess IA marks. Attendance at the seminar is obligatory for all postgraduate students in the program. Student(s) have to give the two presentation/seminar (C1 and C2) during the project work.

Project:

Final-year M.Sc. students are anticipated to dedicate their last semester (4th semester) to working on a project, ideally in a software industry or a research organization (internship). If a student opts to undertake a project within the department, it is recommended to select Research and Development or Department/University-usable projects and carry out the same. The project is to be completed individually, and there will be no pooling of students for a single project. Publication of a paper in an indexed journal or conference is not mandatory as part of the project work; it's optional. However, if a supervisor insists on publication, students are required to comply.

GUIDELINES FOR PROJECT WORKS AND EVALUVATION

PROJECT GUIDELINES

Preamble: Project work is integrated into the M.Sc. curriculum to offer students practical software development experience, focusing on software engineering principles. Throughout the final semester, students engage in all stages of the software development life cycle (SDLC), creating reliable systems. Projects address real-world issues, demanding original, application-oriented solutions. Plagiarism is strictly prohibited, with projects expected to be non-trivial and analytical. Equal team participation is encouraged, and timely completion of all project development activities is required.

GENERAL GUIDELINES

- The Project Work schedule for the fourth semester will be announced beforehand, detailing deadlines for various project milestones. These include submission dates for the Project Proposal, Project Acceptance, Project Synopsis, Problem Analysis Document, System Design Document, Database Design, Detailed Design, Coding and Testing, Final Report, Internal Assessment exams (at least two), and Viva/ Voce assessments.
- Students are encouraged to tackle projects addressing real-life problems, either within their colleges or in industry/research and development settings, as advised by M.Sc. project faculty. For industry projects, faculty must verify their authenticity and originality to ensure their genuine nature.

- At least two internal assessment exams will be held to gauge students' progress during
- Various project stages. These assessments may encompass written tests, document scrutiny, presentations, work demonstrations, group discussions, and viva-voce sessions. This multi-faceted approach ensures an objective evaluation of students' comprehension throughout their project endeavors.

PROJECT VALUATION

External and internal evaluators jointly conduct project assessments in an impartial manner. They define precise evaluation criteria through discussions. Students present their work either through live software demonstrations or, if source code confidentiality is an issue, through verified PPT presentations with company endorsements. The assessment aims to measure students' dedication and comprehension. Evaluators verify adherence to SDLC principles in project reports, taking into account their relevance in context for equitable evaluation. The primary focus is on achieving project goals and documenting SDLC compliance, ensuring thorough assessment while addressing domain-specific nuances for impartiality.

SCHEME OF VALUATION AND MARKS DISTRIBUTION

Particulars		Marks
Internal Assessment		20
Project Report Valuation: Based on the Innovativeness and utility of the project for Industry/Academic or Society (Utility) Related studies about the project (Adequacy), Project plan & implementation-target achieved/output delivered (effectiveness).		50
1	Live Demonstration (Software execution) or Dry runs (Presentation of authentic screenshots or captured videos may be used to walk through complete scenarios)-consistency and completeness	10
2	Question and Answer (Oral only or Oral and written)	20
	Total Marks	100

FORMAT OF PROJECT SYNOPSIS:

Synopsis is a brief outline or general view, as of a subject or written work; an abstract or a summary of the Project Work. It must be as brief (NOT MORE THAN 10 A4 sized paper pages) as is sufficient enough to explain the objective and implementation of the project that the candidate is going to take up.

The write up must adhere to the guidelines and should include the following:

- Title of the Project.
- Introduction, objectives and scope of the Project.
- Tools / Platform, Hardware and Software Requirement specifications.
- Analysis (DFDs at least up to second level, ER Diagrams/ Class Diagrams, Database Design etc., as per the project requirements).
- A complete structure which includes: Number of modules and their description to provide an estimation of the student's effort on the project, Data Structures as per the project requirements for all the modules, Process logic of each module, testing process to be used, reports generation (Mention tentative content of report).
- Whether Industry Defined/Client Defined/User Defined Project? Mention the type. Mention the name and Address of the Industry/Client.
- Limitation of the project.
- Future scope and further enhancement of the project.

GUIDELINES FOR PREPARATION OF DISSERTATION

ORGANISATION OF THE DISSERTATION

The dissertation shall be presented in a number of chapters; starting with Introduction and ending with Conclusion. Each of the chapters will have precise title reflecting the contents of the chapter. A chapter can be subdivided into sections, sub-sections and sub- sub-section so as to present the content discretely and with due emphasis.

The following sequence may be followed in the preparation of the final dissertation report:

- Cover Page (On the hardbound cover)
- Title Page (Inner Cover Page)
- Certificate from the Institute
- Certificate from the Company/Industry/Institution from where the project has been completed
- Declaration Certificate (signed by both Student(s) and guide)
- Acknowledgement
- Table of Contents (with page numbers).
- List of Figures (with figure number, figure titles and page numbers)
- List of Tables with table number, table title and page number.
- Chapters 1, 2………

Sample Chapters:

1. **Introduction**
   a. Introduction of the System

        i. Project Title
        ii. Category
        iii. Overview
    b. Background
        i. Introduction of the Company
        ii. Brief note on Existing System
    c. Objectives of the System
    d. Scope of the System
    e. Structure of the System
    f. System Architecture
    g. End Users
    h. Software/Hardware used for the development
    i. Software/Hardware required for the implementation

2. SRS

    a. Introduction (Brief write-up about SRS)
    b. Overall Description
        i. Product perspective
        ii. Product Functions
        iii. User characteristics.
        iv. General constraints
        v. Assumptions
    c. Special Requirements (Software / Hardware-if any)
    d. Functional requirement.
        i. Module 1
        ii. Module 2
    e. Design Constraints
    f. System Attributes
    g. Other Requirements (if any)

3. System Design (Functional Design)

    a. Introduction (brief write-up about System Design)
    b. Assumptions and Constraints
    c. Functional decomposition
    d. Description of Programs
        i. Context Flow Diagram (CFD)
        ii. Data Flow Diagrams (DFDs–Level 0, Level 1, Level 2)
    e. Description of components
        i. Functional component 1
        ii. Functional component 2

4. Database Design (or Data structure)

    a. Introduction (brief write-up about Database design)
    b. Purpose and scope

c. Table Definition
d. ER diagram

5. **Detailed Design (Logic design of modules)**
   a. Introduction (brief write-up about Database design)
   b. Structure of the software package (structure chart)
   c. Modular decomposition of the System
      i. Module1
         1. Inputs
         2. Procedural details
         3. File I/O interfaces
         4. Outputs
         5. Implementation aspects (if any)
      ii. Module 2
6. **Program code listing**
   a. Database connection
   b. Authorization / Authentication
   c. Data store / retrieval /update
   d. Data validation
   e. Search
   f. Named procedures / functions
   g. Interfacing with external devices (if any)
   h. Passing of parameters
   i. Backup/recovery
   j. Internal documentation
7. **User Interface (Screens and Reports)**
   a. Login
   b. Main Screen / Home page
   c. Menu
   d. Data store / retrieval / update
   e. Validation
   f. View
   g. On screen reports
   h. Data Reports
   i. Alerts
   j. Error messages
8. **Testing**
   a. Introduction (brief write-up about Software Testing)
      i. Unit Testing
      ii. Integrate Testing
      iii. System Testing
   b. Test Reports

- **Conclusion**

- Limitations
- Scope for enhancement (future scope)
- Abbreviations and Acronyms (list)
- Bibliography / References (list in specified format)

Note: Do not include any header or footer in any page of the report. Only page numbers should be mentioned at the bottom center of each page. 'n' copies of dissertation along with soft copy in CD should be prepared by the candidate.

DISSERTATION FORMAT

Paper Quality

The dissertation shall be printed on white bond paper, whiteness 95% or above, weight 70 gram or more per square meter.

Paper Size

The size of the paper shall be standard A4; height 297 mm, width 210 mm.

Type-Setting, Text Processing and Printing

The text shall be printed employing LaserJet or Inkjet printer, the text having been processed using a standard text processor. The standard font shall be Times New Roman of 12 pts with 1.5 line spacing.

Page Format

- Top margin .5"
- Bottom margin .5"
- Left margin 1"
- Right margin .75"

AUXILIARY FORMATS

- **Binding:** The dissertation shall be hard cover bound in leather or rexin.
- **Front Covers:** Full title of dissertation in 6 mm 22-point size font properly centered and positioned at the top. Full name of the candidate in 4.5 mm 15-point size font properly centered at the middle of the page. A 40 mm diameter replica of the college emblem followed by the name of the Department and the year of submission, each in a separate line and properly centered and located at the bottom of the page.
- **Lettering:** All lettering shall be embossed in gold.
- **Bound back:** The degree, the name of the candidate and the year of submission shall also be embossed on the bound (side) in gold.

Prof. Shivanand Gornale
Chairman BOS in Computer Science
RCU, Belagavi

Prof. Basavaraj Padmashali
Dean
Faculty of Science and Technology