Textbooks

1. New York: Wiley. Linz, Peter. An introduction to formal languages and automata. Jones & Bartlett Learning. Seventh Edition.
2. "Introduction to the Theory of Computation" by Michael Sipser, Third Edition.

Reference Books

1. Cohen, Daniel IA, Introduction to computer theory, 2nd Edition.
2. Parkes, Alan P. Introduction to languages, machines and logic: computable languages, abstract machines and formal logic. Springer Science & Business Media, 2012., 2nd Edition

COURSE OUTCOMES (CO):

Upon successful completion of this course, learners will be able to:

CO1: Interpret the role of computational models and finite automata in computer science, recognizing their significance in formal language theory and computational complexity.

CO2: Design and analyze deterministic and non-deterministic finite automata (DFA and NFA), demonstrating an understanding of regular languages and their equivalence with regular expressions.

CO3: Understand and work with context-free languages (CFLs), context-free grammars (CFGs), parse trees, and pushdown automata (PDA), including identifying their properties and limitations.

CO4: Explore and analyze advanced automata and language classes, including context-sensitive languages (CSL), linear bounded automata (LBA), recursive and recursively enumerable languages (REL), and Turing machines (TM), and understand their roles in computational theory.

CO5: Analyze decidability and complexity concepts, including applying reduction techniques and understanding the basics of complexity theory, such as asymptotic notation and the models of deterministic and non-deterministic Turing machines.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	-	3	2	-	-	-	-	-	-	2
2	3	3	3	3	-	2	1	2			2	3
3	2	3	3	3	2	2		2	3	3		3
4	-	3	3	3	1		-	1	1	2	1	3
5	3	2	1	2	-	-	-	-	-	2	-	2

COURSE TITLE: Python Programming

Course Code: MCA24002OE						Examination Scheme	T	P
Total number of Lecture Hours: 28 Total number of Practical Hours: -						External	40	-
						Internal	10	-
Lecture (L):	2	Practical(P):	0	Tutorial (T):	0	Total Credits	2	

Course Objectives

- Students will understand and apply Python variables, operators, data types, and control structures to create basic programs. They will gain the ability to manage program flow using conditional blocks and loops.
- Students will learn to effectively use and manipulate Python data structures, including strings, lists, tuples, and dictionaries. They will also practice slicing and looping techniques for efficient data handling.
- Students will acquire skills in organizing Python code into functions, modules, and packages. They will explore the use of external modules and packages, and apply these techniques to create well-structured and reusable Python projects.

Course Content	TEACHING HOURS
UNIT 1: Python Basics: Variables, Data Types, and Control Structures	14 Hrs.
Understanding Python variables, Python basic Operators, python blocks , Data Types, Declaring and using Numeric data types: int, float, complex Using string data type and string operations Defining list and list slicing Use of Tuple data type: Python Program Flow Control Conditional blocks using if, else and elif Simple for loops in python, For loop using ranges String, list and dictionaries Use of while loops in python Loop manipulation using pass, continue, break and else Programming using Python conditional and loops block	
UNIT 2: Python Functions, Modules, and Packages	14 Hrs.
Python Functions, Modules And Packages, Organizing python codes using functions Organizing python projects into modules, Importing own module as well as external modules Understanding Packages Powerful Lamda function in python, Programming using functions, modules and external packages, Python String, List And Dictionary Manipulations.	

Textbooks

1. Kenneth A. Lambert, "The Fundamentals of Python: First Programs," Cengage Learning, ISBN: 978-1111822705, 1st Edition (2011)
2. David Beazley, Brian K. Jones, "Python Cookbook," O'Reilly Publications, 3rd Edition (2013)

To be effective from year-2024

Reference Books

1. Jake VanderPlas "Python Data Science Handbook" O'Reilly Publications, 1st Edition (2016)
2. David Beazley, "Python Essential Reference (4th Edition) "Addison Wesley, 4th Edition (2009)
3. Vernon L. Ceder," The Quick Python Book, Second Edition", Manning Publications, 3rd Edition (2018)

COURSE OUTCOMES (CO):

CO1: Students will be able to write Python programs that effectively utilize variables, operators, and data types, while employing control structures such as conditional statements and loops to solve computational problems.

CO2: Students will show proficiency in creating and manipulating Python data structures, including strings, lists, tuples, and dictionaries. They will be able to use slicing and looping techniques to efficiently handle data.

CO3: Students will be capable of organizing Python code into functions, modules, and packages. They will demonstrate the ability to import and use external modules, and apply these skills in developing organized, modular, and reusable Python projects.

LEVEL OF CO-PO MAPPING TABLE

	Pos											
Cos	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	-	-	3	-	-	-	2	-	-	3
2	3	3	-	-	3	-	-	-	-	-	-	2
3	2	2	3	-	3	-	-	-	2	-	2	-

COURSE TITLE: Problem Solving with C

Course Code: MCA24002GE	Examination Scheme	T	P
Total number of Lecture Hours: 28 **Total number of Practical Hours: -**	**External**	40	-
	Internal	10	-
Lecture (L): 2 Practical(P): 0 Tutorial (T): 0	**Total Credits**	2	

Course Objectives

- Understand the principles of problem-solving and algorithm development.
- Gain proficiency in the C programming language.
- Develop and implement solutions to computational problems.
- Learn to debug, test, and optimize C programs.

Course Content	TEACHING HOURS
UNIT 1: Introduction to C Programing	**14 Hrs.**
Introduction to algorithms and flowcharts, Overview of programming languages, History and features of C, Setting up the programming environment (IDE/Compiler),Writing and executing the first C program, Structure of a C program, Data types, variables, and constants, Input and output operations (scanf, printf),Operators and expressions, Basic arithmetic operations. Control Structures: Decision-making with if, if-else, and nested if statements, Switch-case statements, Loops: for, while, and do-while loops, Break, Continue, and goto statements.	
UNIT 2: Functions, Arrays, String & pointers in C programming	**14 Hrs.**
Functions in C: Defining and declaring functions, Function arguments and return values, Scope and lifetime of variables (local vs. global), Recursion in **Arrays and Strings:** Introduction to arrays: one-dimensional and two-dimensional arrays, Array operations: traversal, insertion, deletion, Understanding strings, String operations: concatenation, comparison, length calculation, Passing arrays and strings to functions.: **Pointers:** Basics of pointers: declaration, initialization, and usage, malloc, calloc, realloc, and free. **Introduction to file** operations in C, Opening, closing, reading, and writing files.	

Textbooks

1. "Programming in ANSI C" by E. Balagurusamy, 9th Edition (2024), McGraw Hill Education
2. "Let Us C" by Yashavant Kanetkar, 19th Edition (2024), BPB Publications
3. "Data Structures Using C" by Reema Thareja, 4th Edition (2024), Oxford University Press.

Reference Books

1. "C Programming and Data Structures" by P. S. Deshpande and O. G. Kakde, 2nd Edition (2023), Dreamtech Press.
2. "C Programming" by K. R. Venugopal and S. R. Prasad, 3rd Edition (2023), McGraw Hill Education.

COURSE OUTCOMES (CO):

CO1: Understand basic C programming concepts: Master data types, control structures, and functions in C.

CO2: Design and implement algorithms: Develop solutions to computational problems using C.

CO3: Improve problem-solving skills: Apply systematic approaches to solve complex problems.

CO4: Apply C programming to real-world tasks: Use C for practical applications like file handling and data management.

LEVEL OF CO-PO MAPPING TABLE

COs	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	1	1	2	-	-	-	1	2	-	2
2	3	3	2	2	2	-	-	-	2	2	-	2
3	3	3	3	2	2	-	-	-	2	2	2	3
4	3	3	3	2	2	1	1	-	2	2	2	3

Semester III

COURSE TITLE: Java Programming

Course Code: MCA24301CR						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: 56						External	80	40
						Internal	20	10
Lecture (L):	4	Practical (P):	2	Tutorial (T):	-	Total Credits		6

Course Objectives

- Understand the fundamental principles of Java programming language, including its syntax, semantics, and basic constructs.
- Explore object-oriented programming concepts such as classes, inheritance, polymorphism, and interfaces in the context of Java.
- Develop proficiency in handling exceptions and errors using Java's exception handling mechanisms.
- Gain practical experience in utilizing Java's standard library classes and packages for tasks like I/O operations, string manipulation, and multithreading.
- Learn to create graphical user interfaces (GUIs) in Java, employing event-driven programming paradigms and integrating various GUI elements.
- Acquire skills in network programming with Java, including socket programming for communication between distributed systems and applications.

COURSE CONTENT	TEACHING HOURS
UNIT 1: Introduction to Java Programming	**14 Hrs**
Introduction to Java Language: Creation of Java. How Java changed the Internet. Features of Java Language. Evolution of Java. Comparison with other languages like C++.Java Virtual Machine (JVM) and Byte-code. Java Language Overview: Lexical issues – Whitespace, Identifiers, Keywords, Literals, Separators, and Comments. Installing JDK.PATH variable. Java program – Structure, Compilation and Execution. Java Class libraries (System Class).main() method. **Data types, Variables and Arrays:** Primitive Data-types and Typed-Literals. Variables – Declaration, Initialization, Scope and Lifetime. Arrays – Single and Multidimensional. Type Conversion and Expression Promotion. **Operators, Expressions and Control statements:** Arithmetic, Bitwise, Relational, Logical, Assignment. Precedence and Associativity. Selection, Iteration and Jump Statements.	
UNIT 2: Object-Oriented Programming in Java	**14 Hrs**
Class Fundamentals: Class Structure (Variable and Method declaration).Modifiers (Access Modifiers and Other Modifiers).Components of Class, Variable and Method declaration. Constructor and finalize(). Garbage Collection. Passing parameters to methods. Variable hiding. Method overloading. Constructor overloading and chaining. Use of this keyword. Code blocks - Static and non-static. **Inheritance:** Mechanism. Role of Access Modifiers. Method Overriding and Shadowing. Use of super keyword. Polymorphism - Early and Late binding. Abstract Class and Interface. Components of Interface declaration. Implementing Interfaces. **Exception Handling:** Mechanism - Exception-Object, Throwing an Exception, and Exception Handler. Catch or Specify policy. Types of Exception - Checked vs Unchecked, Built-in vs User defined. Catching an Exception - try-catch-finally. Specifying an Exception - throws. Manually throwing an Exception - throw. Custom Exceptions. Chained Exceptions.	
UNIT 3: Advanced Java Concepts	**14 Hrs**

Packages: Creating and Importing Packages. CLASSPATH variable. static import. **Strings:** Mutable and Immutable Strings. Creating Strings. Operations on Strings. **Threads:** Creating Threads in Java. Java Thread Lifecycle. Multithreading in Java: Synchronization and Inter- process communication (IPC) in Threads. **Applet:** Java Applet class Architecture. Working and Lifecycle of Java Applet. Displaying text and animation, and passing parameters to Applet. Embedding Applets in a web page.	
UNIT 4: Java GUI Programming and Networking	**14 Hrs**
Event-Driven Programming: Java 1.1 Event Delegation Model – Source object, Event object and Listener object. Methods associated with Source, Event and Listener objects. Low-level vs Semantic events. Adapter classes, Inner classes, and Anonymous Inner classes. Adding GUI elements to Applet. **I/O Streams:** Byte, Character, Buffered, Data, and Object Streams. Standard Streams. File I/O Basics, Reading and Writing to Files. Serializing Objects. **Networking Classes and Interfaces:** TCP/IP Server Sockets in Java. Developing simple networking applications in Java like File transfer, Chatting, etc.	

Textbooks

1. H. Schildt, Java: The Complete Reference, 13th Edition, Tata McGraw Hill, 2023.

Reference Books

1. E. Balagurusamy, Programming with Java: A Primer, 7th Edition, Tata Mcgraw Hill, 2023.
2. H.M. Dietel and P.J. Dietel, Java: How to Program, 11th Edition, Pearson Education, 2017.
3. K. Sierra and B. Bates, Head First Java (Java 5), 2nd Edition, O'Reilly, 2003.
4. C.S. Horstmann and G. Cornell, Java 2 Vol-1 Fundamentals, 7th Indian Reprint, Pearson Education, 2006.

Lab Manual

Week 1

- Download latest version of Java Development Kit (JDK), preferably JDK8 or above (Please visit https://java.com/en/download/).
- Follow the instructions that appear during the Installation ofJDK8, and set PATH variable to the appropriate directory location as instructed in the lecture.

Week 2

- Write a Java program that displays "hello world!" on the screen.
- Write a Java program that receives two integer numbers via keyboard, does their summation, and displays the result. Ensure that only integer values are processed.
- Write a Java program that prints the season name corresponding to its month number using Ifelse and switch-case statements.
- Write a Java program that sorts (using bubble sort) an integer array using for loop.
- Write a Java program that calculates factorial of a number (inputted via keyboard) recursively.
- Write a Java program that creates a 2D integer array with 5 rows and varying number of columns in each row. Using 'for each' variant of for loop display each element of every row.

Week 3

➢ Write a Java program that creates a Class, namely Student.

- Ensure that Age instance variable of the Class is never accessed directly, and its value is never less than 4 and greater than 40 for any Object of the Class (use methods to validate and assign the value).
- Ensure that the constructor always assigns a unique value to Enrollment_No instance variable for every Object of the Class (use a static class variable for counting objects, say Object_Counter).
- Ensure that when an Object is removed, the Object_Counter is automatically decremented (use finalize()), and whenever required the variable can only be accessed using a method even without an Object reference (make the counter private and use a static method to access it).

➢ Write a Java program in which a Class overloads a method sum(), which takes 2 parameters. The overloaded methods should perform summation of either integer or floating-point values

Week 4

➢ Write a Java program that creates a Class namely A that has a private instance variable and method, a protected instance variable and method, a default instance variable and method, and a public instance variable and method. Create another Class say B that inherits from A.

- Show that all except private members are inherited.
- Show that an inherited instance variable can be shadowed (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
- Show that an inherited method can be overridden (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
- Show that the reference variable of type A or B can't access an overridden method of A in the Object of B.
- Show that the reference variable of type A can access a shadowed data member of A in the Object of B.

Week 5

- Write a Java program that creates a Class in which a method asks the user to input 2 integer values, and calls another member function (say div()) to divide the first inputted number by the second number (by passing them as parameters). Handle an exception that can be raised in div() when the denominator equals zero (use try-catch statement).
- Modify the above Java program so that it also creates a Custom Exception that is thrown by div() when the denominator value is 1 (use throw). Handle the exception. c.
- Modify the above Java program so that the exception-handling in not performed by div() rather it only species all the possible exceptions it may throw (use throws). And, the method that calls div() does the exception handling.

Week 6

➢ Create a Java Package (say pack1) that contains 3 Classes (say A, B and C). Write a Java program that uses this package after setting the CLASSPATH variable. Following scenarios must be considered individually:

- Importing the whole package (all the 3 classes)
- Importing only specific class (say Class A only)

➢ Create another Package (say pack2) that contains same number of classes, and same definition for each class, as that of pack1. Write a Java program that imports all classes from both pack1 and pack2 while ensuring that the name conflicts are not encountered while accessing any of these classes.

Week 7

- Write a Java program to count the number of words in a string that is passed as a command line argument.
- Write a Java program to check whether a string is palindrome or not.
- Write a Java program to count the total number of occurrences of a given character in a string.
- Write a Java program to convert a string to char array.

Week 8

- Write a Java program that creates a Class that extends a Thread class. Create 3 objects of the class, each starting a new thread and each thread displaying "I am Thread: " in an infinite loop. The displayed text must be suffixed by the unique name of the thread.
- Write a Java program that creates a Class that implements interface Runnable, and does the same as the above program.
- Write a Java program to implement a solution for producer-consumer problem using synchronization and inter-process communication in Threads.

Week 9

- Write a Java program that creates a Class that extends an Applet class. The applet is embedded in a web page and is passed 2 numeric parameters. The applet shall display the summation result of the parameters passed.
- Write a Java program that creates a Class that extends an Applet class. The applet simulates a marquee by displaying characters of the message one at a time from right to left across the screen. When the message is fully displayed, the message starts again.
- Write a Java program that creates a Class that extends an Applet class. The applet displays bar chart forthe data passed as parameter. The data includes the number of male and female students enrolled in MCA course.

Week 10

- Write a Java program that creates a Class that extends an Applet class. Add GUI elements to the applet so as to create a simple 2-player tic-tac-toe game.
- Write a Java program that creates a Class that extends an Applet class. Add GUI elements to the applet so as to create a simple calculator.

Week 11

- Write a Java program to open and read a file (filename is passed as command line argument), and displays the number of words in the file?
- Write a Java program to copy a file. The source and destination filenames are passed as command line arguments.

Week 12

- Java program to determine number of bytes written to file using DataOutputStream

- Java program to read text from file from a specified index or skipping byte using FileInputStream

Week 13

- Write a Java program (client) that sends a text message to another Java program (server), which receives and displays it.
- Modify the above Java programs so that each of the two programs is able to send and receive the text messages.

Week 14

- Write a Java program (a client) that opens a connection to https://www.Internic.net website and displays information about www.google.com.
- Write a Java program (Client) that sends a text message to another Java program (Server), and the Server displays an acknowledgement message on receiving it.
- Write a Java program (Client) that sends a text string to another Java program (Server), which receives it and sends back the reverse string of the received string.

COURSE OUTCOMES (CO):

CO1: Master Java Fundamentals: Students would be able to demonstrate proficiency in Java syntax, data types, variables, operators, and control structures. They would be able to implement object-oriented programming concepts including classes, objects, methods, and constructors.

CO2: Handle Exceptions and Utilize Java Libraries: Students would be able to implement effective exception handling strategies to manage errors and ensure program reliability. Utilize Java's standard library for tasks such as I/O operations, string manipulation, and collections.

CO3: Develop GUI Applications: Students can design and develop graphical user interfaces (GUIs) using Java Swing or JavaFX frameworks. Implement event-driven programming to create responsive user interfaces and handle user interactions.

CO4: Implement Networked Applications: Students can develop client-server applications using Java's networking APIs. Utilize TCP/IP protocols for tasks such as file transfer, real-time communication, or remote procedure calls (RPC).

LEVEL OF CO-PO MAPPING TABLE

	Pos											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	1	2	1	-	-	2	1	1	1
2	2	2	1	1	2	1	-	-	2	1	1	1
3	2	2	3	1	2	2	-	-	2	2	2	1
4	2	2	2	1	2	1	-	-	2	1	1	1

COURSE TITLE: Web Programming

Course Code: MCA24302CR			Examination Scheme	T	P
Total number of Lecture Hours: 56			External	80	40
Total number of Practical Hours: 56			Internal	20	10
Lecture (L): 4	Practical (P): 2	Tutorial (T): -	Total Credits		6

Course Objectives:

- Gain a comprehensive understanding of fundamental web technologies, including HTML, and CSS.
- Learn the principles of responsive and accessible web design using CSS and various layout techniques.
- Develop proficiency in JavaScript programming for client-side web development, including DOM manipulation and event handling.
- Acquire skills in server-side scripting using PHP to create dynamic and interactive web applications.
- Understand how to integrate and manage databases within web applications using MySQL.
- Combine client-side and server-side technologies to build complete, functional web applications.

Course Content	TEACHING HOURS
UNIT 1: HTML and XHTML	**14 Hrs**
Introduction to HTML and XHTML: History and evolution of HTML and XHTML, Differences between HTML, Basic structure of an HTML document. HTML Basics: Elements and Attributes, Creating paragraphs, headings, and lists, Working with images, links, and tables, Forms and form controls. Advanced HTML: Semantic HTML5 elements, Multimedia elements: audio and video	
UNIT 2: CSS and Web Design	**14 Hrs**
Introduction to CSS: CSS syntax and selectors, Inline, internal, and external CSS, The cascade and inheritance. Styling Text and Elements: Fonts, text properties, and color, Styling lists, links, and tables, The box model: padding, margin, border Layout Techniques: Positioning elements: static, relative, absolute, and fixed, Flexbox and Grid layout systems, Responsive web design principles, Media queries for different devices.	
UNIT 3: JavaScript	**14 Hrs**
Introduction to JavaScript: History and evolution of JavaScript, JavaScript syntax and data types, Variables, operators, and expressions. JavaScript Basics: Functions and scope, Control structures: loops and conditionals, Objects and arrays, The Document Object Model (DOM). JavaScript and the Web: Event handling, Form validation, Working with JSON, AJAX.	

UNIT 4: PHP and Server-Side Programming	**14 Hrs**
Introduction to PHP: History and features of PHP, Installing and configuring PHP, PHP syntax and data types. PHP Basics: Variables, constants, and operators, Control structures: conditionals and loops, Functions and arrays, Working with forms and user input. PHP and Databases: Connecting to a MySQL database, Performing CRUD operations, Prepared statements and security. Advanced PHP: Sessions and cookies, Error handling and debugging	

Textbooks

4. Learning PHP, MySQL & JavaScript by Robin Nixon, O'Reilly Media 4th Edition (2018).
5. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett 1st Edition (2014)

Reference Books

6. Web Design The complete Reference, Thomas Powell, Tata McGrawHill 2nd Edition (2010)
7. HTML and XHTML The complete Reference, Thomas Powell, Tata McGrawHill 5th Edition (2010)
8. JavaScript 2.0 : The Complete Reference, Second Edition by Thomas Powell and Fritz Schneider 2nd Edition (2004)
9. PHP: The Complete Reference By Steven Holzner, Tata McGrawHill 1st Edition (2008)

Lab Manual

Week 1

- Create a basic HTML document with paragraphs, headings, and lists.
- Add images, links, and tables to an HTML document.
- Create a simple form with various form controls (text inputs, radio buttons, checkboxes, etc.).

Week 2

- Create an HTML document using semantic HTML5 elements.
- Validate an XHTML document and correct any errors.
- Convert an HTML document to XHTML and ensure it follows proper syntax rules.

Week 3

- Create a basic CSS file and link it to an HTML document.
- Apply inline, internal, and external CSS styles to a webpage.
- Use CSS selectors to style different HTML elements.

Week 4

- Style lists, links, and tables using CSS.
- Implement the box model: padding, margin, and border.

Week 5

- Create a simple webpage layout using static, relative, absolute, and fixed positioning.
- Create a responsive webpage using media queries.

Week 6

- Implement CSS animations and transactions on a webpage.
- Write a simple JavaScript program that uses variables, operators, and expressions.

Week 7

- Create JavaScript functions and demonstrate scope.
- Implement control structures (loops and conditionals) in JavaScript.

Week 8

- Create and manipulate JavaScript objects and arrays.
- Use the Document Object Model (DOM) to interact with an HTML document.

Week 9

- Write a JavaScript program to handle events on a webpage.
- Write a JavaScript program that handles errors and debugging.

Week 10

- Write a basic PHP script to output "Hello, World!".
- Create a PHP script that uses variables, constants, and operators.
- Write PHP programs using control structures (conditionals and loops).

Week 11

- Create PHP functions and work with arrays.
- Develop a simple form in PHP and handle user input.
- Connect a PHP script to a MySQL database.

Week 12

- Perform CRUD (Create, read, Update, delete) operations in PHP.
- Implement session management in PHP.

Week 13

- Use cookies in PHP to store user preferences.

- Handle errors and debug a PHP application.

Week 14

- Develop the frontend using HTML, CSS and JavaScript.
- Create the backend using PHP and integrate it with MySQL database.

COURSE OUTCOMES (CO):

CO1: HTML and XHTML Proficiency: Students will be able to create well-structured HTML documents, utilizing semantic elements and multimedia integration.

CO2: CSS Styling and Layout: Students will demonstrate the ability to apply CSS for styling web pages, including advanced techniques like Flexbox and Grid layouts.

CO3: JavaScript Development: Students will be able to write JavaScript code to handle user interactions, manipulate the DOM, and implement AJAX for asynchronous data fetching.

CO4: PHP and Form Handling: Students will be proficient in writing PHP scripts for server-side processing, form handling, and user input validation.

CO5: Database Operations: Students will be able to connect web applications to MySQL databases, perform CRUD operations, and manage data securely.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	2	-	2	1	1	-	1	2	1	1
2	3	2	3	-	3	1	1	-	1	2	1	2
3	3	3	3	2	3	1	1	1	1	2	1	2
4	3	3	3	2	3	1	1	1	1	2	1	2
5	3	3	3	2	3	1	1	1	1	2	1	2

COURSE TITLE: Design and Analysis of Algorithms

Course Code: MCA24303CR						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits		4

Course Objectives

- Gain a solid foundation in algorithms, their analysis, and the growth of functions.
- Apply asymptotic notations and techniques to study the time and space complexity of algorithms.
- Explore and apply methods such as recurrences, the Master Method, and randomized algorithms.
- Utilize divide and conquer, greedy, dynamic programming, backtracking, and branch and bound strategies to solve complex problems.
- Learn about P, NP, NP-hard, and NP-complete problems, and understand the significance of Cook's Theorem.
- Evaluate the need for and implement approximation algorithms for solving complex optimization problems.

Course Content	TEACHING HOURS
UNIT 1: Fundamentals of Algorithm Analysis	**14 Hrs**
Introduction to Algorithms, Analysis of Algorithms, Growth of Functions, Asymptotic notations, Recurrences, Substitution method, Iteration method, Recursion trees The Master Method, Timeand Space Complexity study of some basic algorithms.	
UNIT 2: Advanced Algorithmic Techniques	**14 Hrs**
Randomized Algorithms: Identifying the repeated element, Primality testing, Advantages and Disadvantages. Divide and Conquer Strategy: Binary search, Quick sort, Merge sort Greedy Method, General method, Knapsack problem, Single source shortest paths.	
UNIT 3: Optimization and Search Strategies	**14 Hrs**
Dynamic programming Strategy: All pair shortest paths, Traveling salesman problems. Backtracking Strategy: 8-Queen problem, Sum of subsets, Knapsack problem. Branch and Bound Strategy: Least Cost Branch and Bound, 8-Queen Problem	
UNIT 4: Computational Complexity and Approximation Algorithms	**14 Hrs**
Lower boundary theory, Lower bound theory through reductions, P and NP problems. NP hard and NP complete problems, Cook's Theorem Approximate Algorithms and their need, The vertex Cover Problem, The traveling salesman problem, The subset sum problem	

To be effective from year-2024

Textbooks

1. Horowitz, Sahni, Rajasekaran "Fundamentals of Computer Algorithms",Galgotia Publications, Second Edition, 2018

Reference Books

1. Coremen, Leiserson, Rivest,Stein, "Introduction to Algorithms", PHI, 3rd edition, 2009.
2. Michael T. Goodrich, Roberto Tamassia "Algorithm Design and Applications" , Wiley, 2014

COURSE OUTCOMES (CO):

CO1: Demonstrate the ability to analyze and evaluate the efficiency of algorithms using asymptotic notations and various complexity analysis techniques.
CO2: Solve problems using advanced algorithmic techniques, including recurrences, the Master Method, and randomized algorithms.
CO3: Implement divide and conquer, greedy, and dynamic programming strategies to efficiently solve computational problems like sorting, knapsack, and shortest paths.
CO4: Apply backtracking and branch and bound techniques to tackle complex problems such as the 8-Queen problem and least-cost optimization.
CO5: Gain a deep understanding of computational complexity concepts, including P, NP, NP-hard, and NP-complete problems, and the significance of Cook's Theorem.
CO6: Evaluate the necessity of approximation algorithms and effectively apply them to solve optimization problems like the vertex cover, traveling salesman, and subset sum problems.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	2	2	1	-	2	1	1	1	2
2	3	3	2	2	2	1	-	2	1	1	1	2
3	3	3	3	3	3	2	1	2	1	1	1	2
4	3	3	3	3	3	2	1	2	1	1	1	2
5	3	3	3	3	3	2	1	3	1	1	1	3
6	3	3	3	3	3	2	2	3	1	1	1	3

COURSE TITLE: Ethical Hacking

Course Code: MCA24304DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits		4

Course Objectives

- Introduce students to ethical hacking, distinguishing it from malicious hacking, and covering hacker types, legal and ethical considerations, the hacking process, and essential tools, laying a foundation for advanced learning.
- Teach students network information gathering, scanning, and vulnerability assessment using tools like Nmap and Nessus, emphasizing passive/active reconnaissance and practical application for network security.
- Equip students with practical skills in exploiting vulnerabilities, securing networks, and protecting web applications using Kali Linux and tools like Metasploit.
- Equip students with techniques for covering tracks and maintaining anonymity, including log manipulation and the use of anonymity tools like VPNs and Tor

Course Content	TEACHING HOURS
UNIT 1: Introduction to Ethical Hacking	**14 Hrs.**
Definition and purpose, Differences between ethical hacking and malicious hacking, Importance of ethical hacking in cybersecurity, Types of Hackers (White Hat Hackers, Black Hat Hackers, Grey Hat Hackers), Legal and Ethical Considerations: Laws and regulations, Codes of conduct, Importance of permission and documentation, Ethical Hacking Process: Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks. Setting up your Kali Linux Environment Using a virtual environment, Navigating the Linux command line, Essential Linux Commands.	
UNIT 2: Reconnaissance and Scanning	**14 Hrs.**
Information Gathering: Passive vs. Active Reconnaissance, Tools for information gathering (e.g., Google Dorks, Whois), Network metadata analysis, tools for network metadata analysis, Active information gathering techniques: Nmap, Zenmap, Vulnerability Scanning: Understanding vulnerabilities (SQL Injection, XSS Attack), Tools: Nessus, OpenVAS, Enumeration: Identifying network resources and shares, Tools: Netcat, Nbtstat, Case Studies and Practical Examples: Performing a basic scan using Nmap and analyzing scan results.	
UNIT 3: Executing Vulnerability Assessment and Exploitation Techniques	**14 Hrs.**
Exploiting Vulnerabilities: Exploitation frameworks in Kali Linux, Tools: Metasploit Framework, Password Attacks and Brute force: types of password attacks, Password cracking tools, Best practices and mitigations, Essential resources for understanding password attacks, Wireless networking fundamentals: Types of wireless networks, Wireless network components, Basic Kali Linux commands for wireless Networking, Wireless network vulnerabilities and attacks, Tools for wireless Network exploitation, Defending against wireless attacks.	
UNIT 4: Web Application Attacks, Covering Tracks and Reporting	**14 Hrs.**

Web application security fundamentals, Common web application components, Common web application threats, Understanding HTTP and HTTPS, Web Application Firewalls, scanning for vulnerabilities using Nitko, Brute forcing login forms with Hydra, Exploiting SQL injection with sqlmap, Web application attacks vulnerabilities, Web application exploitation tools and techniques,
Covering Tracks: Importance of covering tracks, Techniques: Log manipulation, clearing logs, spoofing, Anonymity Tools: Proxy servers, VPNs, and Tor, Tools: Proxy-Chains, Tor Browser, Reporting and Documentation: Importance of reporting in ethical hacking, Structure of a penetration testing report, writing an executive summary, and Creating a sample penetration testing report.

Textbooks

3. "Kali Linux for Ethical Hacking: Penetration testing and vulnerability assessment for network security" by Mohamed Atef, First Edition, BPB, 2024.
4. Network Security Assessment: Know Your Network by Chris McNab, 3rd Edition, Oreily, 2017

Reference Books

3. Ethical Hacking and Penetration Testing Guide, by Rafay Baloch, CRC Press, 1e, 2015.

COURSE OUTCOMES (CO):

After completing the course, the student will be able to:
CO1: Understand ethical hacking, its distinction from malicious hacking, hacker types, legal/ethical aspects, and basic command-line/Linux tools.
CO2: Conduct detailed information gathering with passive and active reconnaissance, perform network scanning with tools like Nmap and Nessus, analyze scan results to assess network security, and apply enumeration techniques to uncover network resources and vulnerabilities.
CO3: Apply Kali Linux tools for exploitation, secure networks, and protect web applications by identifying vulnerabilities and implementing defenses.
CO4: Implement anonymity techniques like VPNs and Tor, maintain ethical hacking integrity, and create detailed penetration testing reports with actionable recommendations using tools like Proxy-Chains.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	2	-	2	2	-	3	-	2	-	2
2	3	3	2	-	3	-	-	1	2	-	-	2
3	3	2	2	1	2	2	3	2	2	-	-	2
4	2	2	1	2	3	1	-	2	1	3	-	2

COURSE TITLE: Computer Vision

Course Code: MCA24305DCE	Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -	External	80	-
	Internal	20	-
Lecture (L): 4 Practical (P): - Tutorial (T): -	Total Credits		4

Course Objectives

- Gain a comprehensive understanding of the principles, methodologies, and challenges in computer vision, including image formation, representation, and basic processing techniques.
- Develop the ability to extract meaningful features from images and apply advanced segmentation techniques, preparing for complex tasks like object recognition.
- Learn to apply computer vision algorithms for motion tracking, enabling the analysis and tracking of object movement in real-world video sequences.
- Acquire the skills to perform optical flow analysis, feature matching, and depth estimation, with a focus on 3D reconstruction using multi-camera systems.

Course Content	TEACHING HOURS
UNIT 1: Introduction. Introduction to Computer Vision: Overview, History, and Applications of Computer Vision, Key Challenges, Image Formation and Representation, Basic Image Processing Techniques, Overview of Computer Vision Algorithms.	14 Hrs
UNIT 2: Feature Extraction and Image Segmentation Feature Extraction: Edges (Canny, LOG, DOG), Line Detectors (Hough Transform), Corners (Harris and Hessian Affine), Orientation Histogram (SIFT, SURF, HOG, GLOH), Scale-Space Analysis (Image Pyramids and Gaussian Derivative Filters), Other Filters (Gabor Filters and DWT). Image Segmentation: Region Growing, Edge-Based Approaches to Segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation.	14 Hrs
UNIT 3: Object Motion & Tracking Object Motion & Tracking: Understand Motion Models (Define Object Movement Over Time), Analyze Videos as Sequences of Individual Image Frames, Programmatically Track a Single Point Over Time, implement a Method for Tracking a Set of Unique Features Over Time.	14 Hrs
UNIT 4: Optical Flow & Feature Matching Optical Flow & Feature Matching: Optical Flow (Track a Moving Car Using Optical Flow), Feature Matching (Match Features from One Image Frame to Another), Depth Estimation and Multi-camera Views (Perspective, Binocular Stereopsis, Camera and Epipolar Geometry, Homography, Rectification, DLT, RANSAC, 3-D Reconstruction Framework, Auto-calibration).	14 Hrs

Textbooks:

1. Computer Vision: Algorithms and Applications" by Richard Zaleski, Springer Nature, 2nd Edition, 2022
2. Multiple View Geometry in Computer Vision" by Richard Hartley and Andrew Zisserman, Cambridge University Press, 2nd Edition, 2004
3. Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson Publishing, 4th Edition 2018

Reference Books:

1. Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce, Pearson Publishing, 2nd Edition, 2012

COURSE OUTCOMES (CO):

CO1: The students will be able to Understand and articulate the fundamental principles, challenges, and applications of computer vision, including image formation and basic processing techniques.

CO2: The students will accurately implement and analyze feature extraction techniques like edge and corner detection, as well as orientation histograms, for identifying critical image features.

CO3: The students will be able to apply and evaluate advanced image segmentation methods, including region growing, edge-based approaches, and graph cuts, for accurate image partitioning.

CO4: The students will be able to develop and implement robust object tracking algorithms using motion models and feature tracking techniques to monitor object movement across video frames.

CO5: The students will be able to effectively utilize optical flow techniques for motion analysis and perform precise feature matching across image frames to support depth estimation and 3D reconstruction.

CO6: The students will be able to integrate multi-camera views using principles of stereopsis, camera geometry, and homography to reconstruct 3D scenes and objects with high accuracy.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	1	1	1	2	-	2	-	1	1	1	1
2	1	2	1	1	2	1	2	-	1	2	2	2
3	2	1	3	2	3	2	3	-	2	1	1	1
4	2	1	2	2	2	2	1	-	1	2	2	2
5	1	1	3	3	2	2	2	-	2	1	1	2
6	2	1	3	3	2	2	2	-	2	2	2	1

COURSE TITLE: Enterprise Resource Planning

Course Code: MCA24306DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits		4

Course Objectives

- Understand the evolution, definition, and growth of ERP, including its advantages, various modules, and relevance to different business models, particularly in the context of India.
- Analyze the relationship between ERP and related technologies, such as Business Process Reengineering (BPR), Management Information Systems (MIS), and Supply Chain Management (SCM).
- Evaluate the ERP implementation lifecycle, including planning, system selection, training, data migration, and the roles of consultants, vendors, and employees in successful ERP implementation.
- Assess the post-implementation challenges of ERP systems, the factors influencing their success or failure, and explore emerging trends such as extended ERP systems, CRM, SCM, and web-enabled ERP solutions.

Course Content	TEACHING HOURS
UNIT 1: Introduction to ERP	14 Hrs
Evolution of ERP; . what is ERP?. Reasons for the Growth of ERP; Scenario and Justification of ERP in India; Evaluation of ERP; . Various Modules of ERP; Advantage of ERP.An Overview of Enterprise An Overview of Enterprise;. Integrated Management Information; Business Modeling; ERP for Small Business; ERP for Make to Order Companies; Business Process Mapping for ERP Module Design; . Hardware Environment and its Selection for ERP Implementation	
UNIT 2: ERP and related Technologies	14 Hrs
ERP and Related Technologies; Business Process Reengineering (BPR);. Management Information System (MIS); Executive Information System (EIS); Decision support System (DSS); Supply Chain Management (SCM	
UNIT 3: ERP Implementation	14 Hrs
ERP Implementation: Planning Evaluation and selection of ERP systems - Implementation life cycle - ERP implementation, Methodology and Frame work-Training – Data Migration - People Organization in implementation-Consultants, Vendors and Employees	
UNIT 4: Post Implementation and Emerging Trends	14 Hrs
Post Implementation: Maintenance of ERP- Organizational and Industrial impact; Success and Failure factors of ERP Implementation Emerging Trends on ERP: Extended ERP systems and ERP add-ons -CRM, SCM, Business analytics - Future trends in ERP systems-web enabled.	

A1

To be effective from year-2024

Textbooks

1. Manufacturing Resource Planning (MRP II) with Introduction to ERP; SCM; an CRM by Khalid Sheikh, Publisher: McGraw-Hill
2. ERP and Supply Chain Management by Christian N. Madu, Publisher: CHI.

Reference Books

1. Jagan Nathan Vaman, ERP in Practice, Tata McGraw-Hill, 2008
2. Alexis Leon, Enterprise Resource Planning, second edition, Tata McGraw-Hill, 2008.
3. Mahadeo Jaiswal and Ganesh Vanapalli, ERP Macmillan India, 2009
4. Vinod Kumar Grag and N.K. Venkitakrishnan, ERP- Concepts and Practice, PHI.

COURSE OUTCOMES (CO):

To know the basics of ERP

CO1: To understand the key implementation issues of ERP

CO2: To know the business modules of ERP

CO3: To be aware of some popular products in the area of ERP

CO4: To appreciate the current and future trends in ERP

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	2	2	2	-	-	2	1	1	2	2
2	2	3	2	2	2	-	-	1	1	1	1	2
3	1	2	1	1	2	1	-	1	1	1	1	2
4	1	2	2	2	3	1	-	1	1	1	1	3

COURSE TITLE: Software Quality Assurance

Course Code: MCA24307DCE	Examination Scheme	T	P
Total number of Lecture Hours: 56	External	80	-
Total number of Practical Hours: -	Internal	20	-
Lecture (L): 4 | Practical (P): - | Tutorial (T): -	Total Credits		4

Course Objectives

- Understand the components of the Software Quality Assurance System, including Pre-Project Software Quality Components, Contract Review, and Development and Quality Plans.
- Apply various quality engineering tools and techniques, such as the Seven Basic Quality Tools, Statistical Process Control, and Failure Mode and Effect Analysis (FMEA).
- Analyze different software quality standards and models, including ISO 9000 series, CMMI, Six Sigma, and the integration of AI techniques in software testing and quality assurance.
- Implement Scrum in an organization, understanding the steps for transitioning to Scrum, Scrum artifacts, and the use of Agile project management tools like JIRA and Trello.

Course Content	TEACHING HOURS
UNIT I: Software Quality	**14 Hrs**
Definition of Software Quality, Quality Planning, Quality system ,Quality Control vs Quality Assurance , Product life cycle , Project life cycle models. The Software Quality Challenge, Software Quality Factors, Components of the Software Quality Assurance System. Pre-Project Software Quality Components, Contract Review, Development and Quality Plans	
UNIT II: Software Quality Engineering Tools And Techniques	**14 Hrs**
Supporting quality activities- Metrics, Reviews, SCM , Software quality assurance and risk management, Seven basic Quality tools, Checklist, Pareto diagram, Cause and effect diagram, Run chart, Histogram, Control chart, Scatter diagram, Poka Yoke, Statistical process control ,Failure Mode and Effect Analysis, Quality Function deployment, Continuous improvement	
UNIT III: Quality Assurance Models and AI Assurance	**14 Hrs**
Software Quality Standards, ISO 9000 series, CMM, CMMI, P-CMM , Six Sigma, Malcolm Baldrige Quality, Introduction to AI in Software Quality Assurance-Definition and Importance of AI Assurance, Overview of AI in Software Development and Testing, Benefits and Challenges of Integrating AI in QA, Overview of AI Techniques in Software Testing, Automated Test Case Generation-Techniques for Automated Test Case Generation, Ensuring AI Model Quality-Verification and Validation of AI Models, Testing AI Models for Accuracy, Robustness, and Fairness	

Unit IV: Scrum and Agile Model	14 Hrs
Introduction to Agile-Definition and History of Agile, Principles of Agile Manifesto, Benefits of Agile Methodology, Overview of Scrum Framework-Definition of Scrum, Scrum Values and Principles, Scrum Artifacts-Product Backlog, Sprint Backlog, Increment, Agile Project Management Tools-Introduction to Tools (e.g., JIRA, Trello, Asana), Implementing Scrum in an Organization-Steps to Transition to Scrum	

Textbooks:

1. Galin, D. (2018). Software Quality Assurance: From Theory to Implementation (2nd ed.). Pearson.

2. Godbole, N. S. (2017). Software Quality Assurance: Principles and Practice (1st ed.). Alpha Science International Ltd.

Reference Books:

1. Cohn, M. (2006). Agile Estimating and Planning. Prentice Hall.
2. Partridge, D. (1992). Artificial Intelligence in Software Engineering. Routledge.
3. Black, R. (2002). Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing (2nd ed.). Wiley.
4. Jones, C. (2017). Applied Software Measurement: Global Analysis of Productivity and Quality (3rd ed.). Addison-Wesley Professional.

COURSE OUTCOMES (CO):

CO1: Understand and differentiate between software quality control and software quality assurance, and describe the components of a comprehensive Software Quality Assurance System.

CO2: Apply quality engineering tools and techniques such as checklists, Pareto diagrams, control charts, and Failure Mode and Effect Analysis (FMEA) to enhance software quality and manage risks.

CO3: Analyze and implement various software quality standards and models, including ISO 9000 series and CMMI, and integrate AI techniques in software testing and quality assurance processes.

CO4: Implement and manage Agile methodologies, specifically the Scrum framework, including its principles, artifacts, and project management tools, to enhance software development processes.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	2	2	2	2	1	2	2	1	-	2
2	3	3	3	2	3	2	2	2	2	2	1	3
3	3	3	3	3	3	3	2	3	3	2	2	3
4	3	3	3	2	3	2	2	2	3	3	3	3