Post Graduate Department of Computer Sciences, The University of Kashmir, Srinagar-190006

Proposed Scheme & Syllabus of MCA Programme

Eligibilty for 2- year MCA Programme

Passed BCA/B.Sc./B.E/B.Tech/B. Com/B. A/B.Voc. etc. with at least 50% marks in case of general category and 45% marks in case of reserved category shall be eligible to apply for admission to the MCA programme.

Provided that the candidates with no mathematical background at UG / +2 shall be required to pass the compulsory bridge course/s in Mathematics and Computer related subjects.

To be effective from year-2024

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: To prepare students to get employment, profession and/or to pursue post-graduation and research in Computer Applications discipline in particular and allied Computer Science fields in general.

PEO2: To prepare students to identify and analyze problems in the computing perspective an develop computer applications solutions using an iterative approach that involves defining, designing, quantifying, implementing, testing, deploying and review of the solution to the problem..

PEO3: To prepare students to plan, organize, schedule, execute and communicate effectively as an individual, a team member or a leader in problem solving environment.

PEO4: To provide to students, an academic environment that makes them aware of excellence in field of Computer Sciences in general and enables them to understand significance of lifelong learning in global perspective.

PROGRAM OUTCOMES (POs)

Computer Applications Masters will be able to:

1. Computing knowledge: Apply the knowledge of computing, mathematics and engineering fundamentals to the solution of complex software engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex computing problems reaching substantiated conclusions using first principles of computing, analytics, algorithms and software engineering sciences.

3. Design/development of solutions: Design computer application solution for complex computing problems and design software or processes that meet the specified needs with appropriate consideration for the public ethics, health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern software engineering and development tools including prediction and modeling to complex computing activities with an understanding of the limitations.

6. The software engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, ethical, legal and cultural issues and the consequent responsibilities relevant to the professional software development practice.

To be effective from year-2024

7. Environment and sustainability: Understand the impact of the professional software engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the software engineering practice with most important stress on privacy.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex problem solving activities with the software engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the software engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

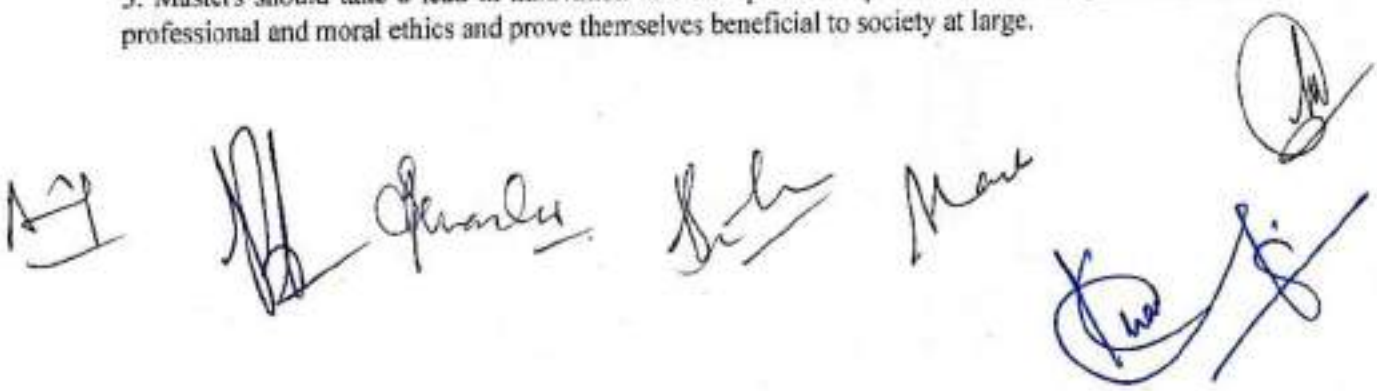
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes:

1. Masters should be creative, imaginative and proficient software engineers employable to serve in the industry, government and allied services.

2. Masters should be able to advance in academic and research pursuits in computing and allied disciplines.

3. Masters should take a lead in innovation and entrepreneurship activities with high standards of professional and moral ethics and prove themselves beneficial to society at large.



To be effective from year-2024

SEMESTER I

Subject Code	Subject Name	Subject Category	Hours/Week			Credit Units
			L	T	P	
Core Courses (16 Credit Units)						
MCA24101CR	Programming with C++	Core	4	0	4	6
MCA24102CR	Database Systems	Core	4	0	4	6
MCA24103CR	Discrete Mathematics	Core	4	0	0	4
Discipline Centric Electives (8 Credit Units)						
MCA24104DCE	Computer Networks	DCE	4	0	0	4
MCA24105DCE	Computer Graphics	DCE	4	0	0	4
MCA24106DCE	Management Information System	DCE	4	0	0	4
MCA24107DCE	Software Engineering	DCE	4	0	0	4
MCA24108DCE	Artificial Intelligence	DCE	4	0	0	4
MCA24109DCE	Block Chain Technologies	DCE	4	0	0	4
MCA24110DCE	Computer Architecture & ALP	DCE	4	0	0	4
OE/GE (2 Credit Units) For Students of Other Departments						
MCA24001OE	Digital and Technological Solutions	OE	2	0	0	2
MCA24001GE	Data Processing using Spreadsheets	GE	2	0	0	2

SEMESTER II

Subject Code	Subject Name	Subject Category	Hours/Week			Credit Units
			L	T	P	
Core Courses (16 Credit Units)						
MCA24201CR	Data Structures Using C++	Core	4	0	4	6
MCA24202CR	Data Science with Python	Core	4	0	4	6
MCA24203CR	Operating Systems	Core	4	0	0	4
Discipline Centric Electives (8 Credit Units)						
MCA24204DCE	Cryptography and Network Security	DCE	4	0	0	4
MCA24205DCE	Digital Image Processing	DCE	4	0	0	4
MCA24206DCE	Decision Support Systems	DCE	4	0	0	4
MCA24207DCE	Software Project Management	DCE	4	0	0	4
MCA24208DCE	Machine Learning	DCE	4	0	0	4
MCA24209DCE	Cloud Computing	DCE	4	0	0	4
MCA24210DCE	Linux Programming	DCE	4	0	0	4
MCA24211DCE	Theory of Computation					
OE/GE (2 Credit Units) For Students of Other Departments						
MCA24002OE	Python Programming	OE	2	0	0	2
MCA24002GE	Problem Solving with C	GE	2	0	0	2

To be effective from year-2024

SEMESTER III

Subject Code	Subject Name	Subject Category	Hours/Week			Credit Units
			L	T	P	
Core Courses (16 Credit Units)						
MCA24301CR	Java Programming	Core	4	0	4	6
MCA24302CR	Web Programming	Core	4	0	4	6
MCA24303CR	Design and Analysis of Algorithms	Core	4	0	0	4
Discipline Centric Electives (8 Credit Units)						
MCA24304DCE	Ethical Hacking	DCE	4	0	0	4
MCA24305DCE	Computer Vision	DCE	4	0	0	4
MCA24306DCE	Enterprise Resource Planning	DCE	4	0	0	4
MCA24307DCE	Software Quality Assurance	DCE	4	0	0	4
MCA24308DCE	Deep Learning	DCE	4	0	0	4
MCA24309DCE	Internet of Things (IoT)	DCE	4	0	0	4
MCA24310DCE	Cyber Security and Digital Forensics	DCE	4	0	0	4
OE/GE (2 Credit Units) For Students of Other Departments						
MCA24003OE	Web Development	OE	2	0	0	2
MCA24003GE	Data Analytics	GE	2	0	0	2

SEMESTER IV

Subject Code	Subject Name	Subject Category	Hours/Week			Credit Units
			L	T	P	
Core Courses (16 Credit Units)						
MCA24401CR	Project: Problem Identification & Analysis	Core	0	8	0	8
MCA24402CR	Project: Dissertation	Core	0	8	0	8
Discipline Centric Electives (8 Credit Units)						
MCA24403DCE	Project: Software Development	DCE	0	8	0	8
MCA24404DCE	Project: Research Component	DCE	0	8	0	8
OE/GE (2 Credit Units) For Students of Other Departments						
MCA24004OE	Latex	OE	2	0	0	2
MCA24004GE	AI Tools	GE	2	0	0	2

To be effective from year-2024

Semester I

To be effective from year-2024

COURSE TITLE: Programming with C++								
Course Code: MCA24101CR						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

- To introduce students to the basic data types, variables, constants, and literals in C programming and to teach them how to use arithmetic, relational, logical, and bitwise operators.
- To teach students the various control structures, such as if-else, switch statements, and loops (while, do-while, for), and how to effectively use them to control the flow of a program.
- To enable students to understand and work with one-dimensional, two-dimensional, and multi-dimensional arrays, and to manipulate strings and character arrays using standard library functions.
- To develop students' ability to write functions, including prototypes and parameter passing, and to understand storage classes and identifier visibility. To teach recursive functions and their applications.
- To introduce students to advanced topics such as command-line arguments, file processing, structures and unions, and pointers. To explain the scope, lifetime, and multi-file programming.
- To provide a foundation in object-oriented programming with a focus on classes and objects, access specifiers, constructors, destructors, inheritance, polymorphism, and templates in C++. To introduce the concepts of abstraction, encapsulation, and exception handling.

Course Content	TEACHING HOURS
Unit I: Fundamentals of C Programming	**-14 Hrs**
Data Types, Identifiers, Varisbles Constants and Literals. Arithmetic Relational Logical and Bitwise. Basic input/output statements, Control structures: if-else statement, Nested if statement, Switch statement Loops: while loop, do while, for loop, Nested loops. Arrays: Declaration; initialization; 2-dimensional and 3-dimensional array, passing array to function, Strings and String functions, and character arrays. Functions; prototype, passing parameters, storage classes, identifier visibility, Recursive functions	
Unit II: Advanced C Programming Techniques	**-14 Hrs**
Command-line arguments. Variable scope, lifetime. Multi-file programming, Introduction to macros. File processing in C. Structures and unions: syntax and use, members, structures as function arguments passing structures and their arrays as arguments Pointers: variables, pointers and arrays, pointers to pointers, strings, pointer arithmetic, portability issues, pointers to functions, void pointers, pointer to structure. Introduction to object oriented programming, Abstraction, Encapsulation	
Unit III: Introduction to Object-Oriented Programming in C++	**-14 Hrs**
Introduction to classes and objects; Access specifiers, Constructor; destructor; Function overloading; Operator overloading; friend functions; Use of call-by-reference for efficiency. Copy constructor. Inheritance: Single, Multiple, and Multilevel Inheritance ,Virtual functions and Polymorphism/Dynamic binding vs Static binding; Virtual Destructors.	
Unit IV: Advanced Object-Oriented Programming Concepts	**- 14Hrs**

To be effective from year-2024

Pure virtual function; concrete implementation of virtual functions, Templates: Function Templates, Class Templates, Member Function Template and Template Arguments, namespaces, Exception Handling Concepts, Input and Output: Streams classes, Stream Errors, Disk File I/O with streams.	

Textbooks

1. Herbert Schildt , "C++ The complete Reference", 4th Edition, 2017

Reference Books

1. Brian W. Kernighan / Dennis Ritchie, "The C Programming Language", 2nd Edition, 2015
2. Bjarne Stroustrup, "The C++ programming language", 4th Edition, 2022
3. E.Balagurusamy , "Object Oriented Programming with C++" 8th Edition, 2020
4. Reema Thareja, "Programming in C 3e", 3rd edition, 2023
5. E. Balaguruswamy, "Programming in ANSI C", 7th Edition, 2017
6. S.K.Srivastava/Deepali Srivastava, "C In Depth", 2009

Lab Manual: Programming with C++

Week 1

- Write a program to demonstrate the use of Output statements that draws any object of your choice e.g. Christmas Tree using '*'
- Write a program that reads in a month number and outputs the month name.
- Write a program that demonstrate the use of various input statements like getchar(), getch(), scanf().
- Write a program to demonstrate the overflow and underflow of various datatype and their resolution?

Week 2

- Write a program to demonstrate the precedence of various operators.
- Write a program to generate a sequence of numbers in both ascending and descending order.
- Write a program to generate pascals triangle.
- Write a program to reverse the digits of a given number. For example, the number 9876 should be returned as 6789.

Week 3

- Write a program to convert an amount (upto billion) in figures to equivalent amount in words.
- Write a program to find sum of all prime numbers between 100 and 500.
- Create a one dimensional array of characters and store a string inside it by reading from standard input.
- Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number. Print the number and its frequency in a tabular form.

Week 4

- Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.
- Write a program which will arrange the positive and negative numbers in a one-dimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers. Example: Original array contains: 10-15,1,3,-2,0,- 2,-3,2,-9 Modified array: 10,1,3,0,2-15,-2,-2,-3,-9

- Write a program to compute addition multiplication and transpose of a 2-D array.
- Implement a program which uses multiple files for holding multiple functions which are compiled separately, linked together and called by main(). Use static and extern variables in these files.

Week 5

- Implement a function which receiver a pointer to a Student struct and sets the values of its fields.
- Write a program which takes five arguments on command line, opens a file and writes one argument per line in that file and closes the file.
- Write a program which creates Student (struct) objects using malloc and stores their pointers in an array. It must free the objects after printing their contents.
- Write a function char* stuff(char* s1,char* s2,int sp, intrp) to stuff string s2 in string s1 at position sp, replacing rp number of characters (rp may be zero).

Week 6

- Write a program to input name, address and telephone number of 'n' persons (n<=20). Sort according to the name as a primary key and address as the secondary key. Print the sorted telephone directory.
- Write a program to find the number of occurrences of a word in a sentence?
- Write a program to concatenate two strings without using the inbuilt function?
- Write a program to check if two strings are same or not?
- Write a program to check whether a string is a palindrome or not?
- Write a program to find the number of vowels and consonants in a sentence?

Week 7

- Write a program that reverse the contents of a string.
- Write a program to demonstrate the array indexing using pointers.
- Write a program to pass a pointer to a structure as a parameter to a function and return back a pointer to structure to the calling function after modifying the members of the structure?

Week 8

- Write a program to demonstrate the use of pointer to a pointer.
- Write a program to demonstrate the use of pointer to a function.
- Write a program to demonstrate the swapping the fields of two structures using pointers?
- Write a program in C++ to define class complex which having two data members viz real and imaginary part?
- Write a program in C++ to define class Person which having multiple data members for storing the different details of the person e.g. name,age, address, height etc.

Week 9

- Write a program to instantiate the objects of the class person and class complex?
- Write a C++ program to add member function that displays the contents of class person and class complex?
- Write a C++ program to demonstrate the use of scope resolution operator?
- Write a program in C++ which creates objects of Student class using default, overloaded and copy constructors.

Week 10

- Write a program to demonstrate the use of different access specifiers.
- Write a C++ program to demonstrate the use of inline, friend functions and this keyword.
- Write a C++ program to show the use of destructors.
- Write a program in C++ demonstrates the use of function overloading.
- Write a C++ program to overload the '+' operator so that it can add two matrices.

Week 11

- Write a C++ program to overload the assignment operator.
- Write a C++ program to overload comparison operator operator== and operator! =
- Write a C++ program to overload the unary operator.
- Write a program in C++ which creates a single-inheritance hierarchy of Person, Employee and Teacher classes and creates instances of each class using new and stores them in an array of Person *

Week 12

- Write a program in C++ which creates a multiple-inheritance hierarchy of Teacher classes derived from both Person, Employee classes. Each class must implement a Show () member function and utilize scope-resolution operator.
- Write a program in C++ demonstrates the concept of function overriding?
- Write a C++ program to show inheritance using different levels?
- Write a C++ program to demonstrate the concepts of abstract class and inner class?

Week 13

- Write a C++ program to demonstrate the use of virtual functions and polymorphism?
- Write a C++ program to demonstrate the use of pure virtual functions and virtual destructors? Write a C++ program to swap data using function templates.
- Write a C++ program to create a simple calculator which can add, subtract, multiply and divide two numbers using class template.

Week 14

- Write a C++ program to demonstrate the concept of exception handling.
- Write a C++ program to create a custom exception. Define a class with appropriate data members and member functions which opens an input and output file, checks each one for being open, and then reads name, age, salary of a person from the input file and stores the information in an object, increases the salary by a bonus of 10% and then writes the person object to the output file. It continues until the input stream is no longer good.

COURSE OUTCOMES (CO):

CO1: Students will demonstrate proficiency in using basic data types, control structures, and input/output statements to develop efficient C programs.
CO2: Students will develop complex programs involving arrays, strings, and functions, including recursive functions and multi-dimensional arrays.
CO3: Students will apply advanced C programming concepts such as command-line arguments, file processing, and the use of pointers, structures, and unions to solve real-world problems.

CO4: students will implement object-oriented programming principles in C++, including the creation and management of classes and objects, and applying inheritance, polymorphism, and virtual functions.
CO5: Students will be able to utilize function and class templates, namespaces, and exception handling to enhance program robustness and reusability.
CO6: Students will be able to perform input and output operations using stream classes, handle stream errors, and manage disk file I/O operations effectively.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Database Systems								
Course Code: MCA24102CR						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

- Grasp the basic concepts of data, information, and knowledge, and the need for and evolution of databases and DBMS. Analyze the characteristics, advantages, and disadvantages of the DBMS approach.
- Describe data models, schemas, and instances, and compare various database models. Understand the Three Schema Architecture, data independence, database languages, interfaces, and DBMS classifications.
- Gain an overview of data modeling and create entity-relationship (ER) models to represent data structures and relationships effectively.
- Understand the basic concepts, characteristics, and constraints of the relational data model. Apply relational algebra operations, including unary, set theory, and binary operations, to manipulate relational data.
- Apply the criteria for good database design. Use functional dependencies and normalization techniques (1NF, 2NF, 3NF, BCNF) to design efficient and reliable database schemas that ensure data integrity and minimize redundancy.
- Learn SQL syntax and functionalities, including data definition, manipulation, and transaction control. Handle constraints, joins, views, synonyms, indexes, subqueries, and locks in SQL. Understand the basics of transaction processing, concurrency control, schedules, serializability, and recovery mechanisms to ensure database consistency and reliability.

Course Content	TEACHING HOURS
UNIT 1: Introduction to Database Systems	**14 Hrs**
Introduction to Data, Information and Knowledge. Database basics – Need and evolution, Database andDBMS. Characteristics of Database Approach, Advantages and disadvantages of DBMS Approach. Database System Concepts and Architecture – Data Models, Schemas, and Instances. Database Modelsand Comparison Three Schema Architecture and Data Independence. Database Languages and Interfaces. DBMS architectures. DBMS Classification. Data Modeling: Overview of Data Modeling, Entity-Relationship (ER) Modeling.	
UNIT 2: Relational Data Model and Database Design	**14 Hrs**
Relational Data Model –Basic Concepts and Characteristics, Model Notation, Model Constraints and Database Schemas, Constraint Violations. Relational Algebra – basic concepts, Unary Relational Operations, Algebra Operations from Set Theory,Binary Operations, Additional Relational Operations Criterion for Good Database Design. Database Design through Functional Dependencies & Normalization: Functional Dependencies, Lossless Join, Normal Forms: 1NF, 2NF, 3NF, BCNF.	
UNIT 3: SQL and Advanced Data Definition	**14 Hrs**

Introduction to SQL, Data Types, Data Definition Language, Data Manipulation Language, Specifying Constraints in SQL, Transaction Control Language, SQL Functions, Set Operators and Joins, View, Synonym and Index, Sub Queries and Database Objects, Locks and SQL Formatting Commands.	
UNIT 4: Transaction Processing and Database Recovery	**14 Hrs**
Transaction Processing –Transaction Processing Basics, Concurrency Control, Transaction and SystemsConcepts, Desirable properties of Transactions. Characterizing Schedules and Recoverability, Schedules and Serializability. Concurrency Control - TwoPhase Locking, Timestamp Ordering. Database Recovery – Concepts, Transaction Rollback, Recovery based on Deferred and Immediate Update, Shadow Paging	

Textbooks

1. Elmasri and Navathe, Fundamentals of Database Systems, 7/e, Pearson, 2017

Reference Books

1. Silberschatz, Korth, & Sudarshan, Database System Concepts, McGraw-Hill. 7/e, 2011.
2. Bayross I. SQL, PI/SQL: The Programming Language of Oracle, BPB Publications, 2009
3. Michael J. Hernandez, Database Design for Mere Mortals®: A Hands-on Guide to Relational Database Design, Third Edition, Addison-Wesley Professional, 2013

Lab Manual

Week 1

- List various users, functions and constraints of the database system for Library Management.
- List various users, functions and constraints of the database system for Banking System.

Week 2

- Identify the various tables and draw a diagrammatic schema to represent the database of Library management system.
- Identify the various tables and draw a diagrammatic schema to represent the database of University system.

Week 3

- Draw ER Model for the database of Library management system.
- Draw ER Model for the database of University management system.

Week 4

- Consider the following schema: Suppliers (sid, sname, address) Parts (pid, pname, color) Catalog (sid, pid, cost)

Write relational algebra queries to

- Find the name of suppliers who supply some red parts.
- Find the sids of suppliers who supply some red or green parts.
- Find the sids of suppliers who supply some red part or are at Srinagar.
- Find the sids of suppliers who supply some red and some green part.
- Find the sids of suppliers who supply every part.
- Find the sids of suppliers who supply every red part.
- Find the sids of suppliers who supply every red or green part.

Week 5

- Consider a schema R(A,B,C,D) and functional dependencies A->B and C->D. Check the decomposition of R into R1(AB) and R2(CD) for lossless join and dependency preservation.
- R(A,B,C,D) is a relation. Which of the following does not have a lossless join, dependency preserving BCNF decomposition?

1. A->B, B->CD 2. A->B, B->C, C->D
3. AB->C, C->AD 4. A ->BCD

Week 6

- Using a sample schema and data, demonstrate the use of 1NF, 2NF, 3NF and BCNF.

Week 7

- CreatetableStudentwithfollowingattributesandperformthefollowingoperations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS	ST_TELNO
Date Type	Number	Varchar	Char	Varchar2
Size	6	30	35	15

- AddnewattributesCity,Street,CountrywithDatatypeVarcharandlength30?
- Modifyfield ST_ROLLNOand change thesize to5?
- RemovecolumnST_ADDRESS?
- DescribetheTableStudent?
- DropTableStudent?
- CopyStructureof onetableto another

Week 8

- Create Users user1, user2,user3 and perform the following operations
  - Grant Session Privilege to the newly created users?
  - Grant privileges for creating and manipulation tables?
  - Grant data manipulation privileges to various users on tables?
  - Grant privileges with grant option.
  - Revoke privileges.

Week 9

- Create Object ADDRESS and use the object in a Table DDL?
- Create table Student with following attributes and perform the following operations.

Attribute Name	ST_ROLLNO	ST_NAME	ST_STREET	ST_CITY	ST_State	ST_Country	DTE_REG
Date Type	Number	Varchar	Char	Char	Varchar2	Varchar2	Date
Size	6	30	35	30	30	30	

  - Insert 10recordsinthetable.
  - Perform various Project Operations using Select Query.
  - Perform various restrict operations using Select Query.
  - Update records in the table.
  - Delete records in the table.
  - Create another table with same structure as existing table without copying the data.
  - Create another table along with the structure and data from existing table.

Week 10

- Create table Student with ST_ADDRESS as Object Type with following attributes and

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS				DT_REG
			ST_STRET	ST_CITY	ST_State	ST_Country	
DateT ype	Number	Varchar	Char	Char	Varchar 2	Varchar 2	Date
Size	6	30	35	30	30	30	

  - Insert 10 records.
  - Perform various Project Operations using Select Query.
  - Perform various restrict operations using Select Query.
  - Update records in the table
  - Delete records in the table

- Create table STUDENT with following attributes and perform the following operations?

Attribute Name	ST_ROLLNO	ST_NAME	ST_STREET	ST_CITY	ST_State	ST_Country	DTE_REG	
Date Type	Number	Varchar	Char	Char	Varchar2	Varchar2	Date	
Size	6	30	35	30	30	30		

- Insert 10 records in the table.
- Perform various Project Operations using Select Query.
- Perform various restrict operations using Select Query using various arithmetic and Logical Operators like
  - Less Than
  - Greater Than
  - Less Than or Equal to
  - Greater Than or Equal To
  - Equal to
  - Not Equal To
  - Perform restrict operations using various datatypes like numeric,Characters,Date.
  - Perform Update operations using various Arithmetic and Logical Operators on Table STUDENT
- Perform Delete operations using various Arithmetic and Logical Operators on Table STUDENT
- Use Insert and Select Commands together with Arithmetic and Logical Operators.

Week 11

- Perform following Transaction Control Operations on the above table
  - Perform various data manipulation operations the table .
  - Create Five Savepoints from S1 to S5.
  - Rollback to Various savepoints and observe the changes in the table.
  - Perform various DDL operations the table and observe its effect on Savepoint and Rollback on the table.
  - Try to abnormally terminate the application to observe whether data is saved or not.
  - Use Commit and Commit Work commands to save the data permanently.
- Create table STUDENT with following attributes and perform various DML operations to verify domain constraint

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	NOTNull	NotNULL	NOTNULL

Week 12

- Create table STUDENT with following attributes and perform various DML operations to verify Validity Integrity.

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRESS
Date Type	Number	Varchar2	Varchar
Size	6	30	35

Constraint	CHECK (ROLLNO >20001 and ROLLNO<30001	NotNULL	NOTNULL

- Create table STUDENT with following attributes and perform various DML operations to verify Entity Integrity using Primary and Unique Keys?

Attribute Name	ST_ROLLNO	ST_NAME	ST_ADDRSS
Date Type	Number	Varchar2	Varchar
Size	6	30	35
Constraint	Primary/UniqueKeys	NotNULL	NOTNULL

Week 13

- Create table STUDENT with following attributes and perform various DML operations to verify Referential Integrity using given tables (employee and department)?

Attribute Name	EMP_ID	EMP_NAME	ST_ADDRESS	DEPT_ID
Date Type	Number	Varchar2	Varchar	Number
Size	6	30	35	4
Constraint	PrimaryKey	NotNULL	NOTNULL	Foreign Key

Attribute Name	DID	NAME	Address
Date Type	Number	Varchar2	Varchar
Size	4	30	100
Constraint	Primary Key	NotNULL	NOTNULL

Week 14

- Write SQL queries to demonstrate use of Join and various SQL functions

COURSE OUTCOMES (CO):

CO1: Demonstrate the ability to understand the fundamentals of data, information, and knowledge. Evaluate the need, evolution, and characteristics of databases and DBMS, including their advantages and disadvantages.

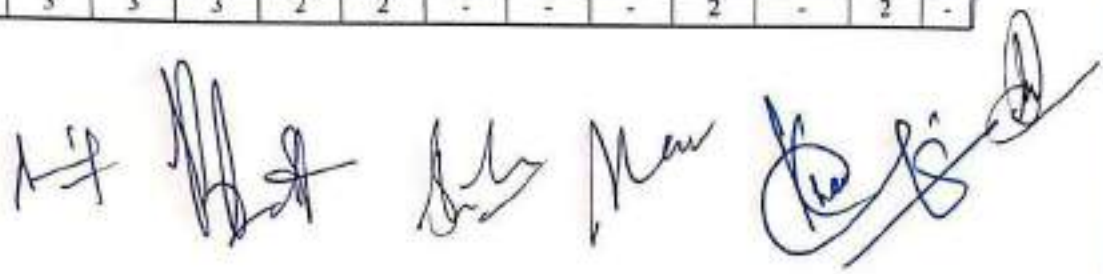
CO2: Describe and apply various database system concepts and architectures, including data models, schemas, instances, and the Three Schema Architecture. Understand and use database languages, interfaces, and DBMS classifications.

CO3: Create effective data models using entity-relationship (ER) modeling. Apply relational data model principles and relational algebra operations. Design and normalize database schemas using functional dependencies and various normal forms (1NF, 2NF, 3NF, BCNF).

CO4: Utilize SQL for defining, manipulating, and controlling data. Handle constraints, joins, views, subqueries, and database objects. Understand and apply transaction processing concepts, concurrency control mechanisms, and database recovery techniques to ensure database consistency and reliability.

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	3	2	2	-	-	-	-	-	2	1	-	-
3	3	3	3	-	2	-	-	-	2	-	2	-
4	3	3	3	2	2	-	-	-	2	-	2	-



COURSE TITLE: Discrete Mathematics								
Course Code: MCA24103CR						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

- Understand and apply fundamental concepts of propositional logic, truth tables, and logical equivalence.
- Demonstrate proficiency in handling predicates, quantifiers, and operations on sets, including cardinality.
- Utilize various methods of proof, including direct proof, indirect proof, and mathematical induction, to solve problems and prove the correctness of algorithms.
- Apply counting techniques such as permutations, combinations, and the Pigeonhole Principle to solve problems in discrete mathematics
- Analyze and apply principles of discrete probability, including advanced counting techniques like the inclusion-exclusion principle and solving recurrence relations.
- Interpret and analyse relations, digraphs, and basic graph theory concepts, including connectivity, paths, circuits, and graph colouring, using appropriate mathematical tools and representations.

Course Content	TEACHING HOURS
UNIT 1: Foundations of Discrete Mathematics	**14 Hrs**
Proposition, Logic, Truth tables, Propositional Equivalence, Logical Equivalence. Predicates and Quantifiers; Sets: operations on sets, Computer representation of sets. Cardinality of a Set Functions: Domain, Range, One-to-One, Onto, Inverses and Composition. Sequences and summations, Growth of functions. Methods of Proof: Direct Proof, Indirect Proof, Mathematical Induction for proving algorithms; Counting techniques – Permutations, Combinations, The Pigeonhole Principle.	
UNIT 2: Probability, Counting, and Relations	**14 Hrs**
Discrete Probability, Advanced Counting Techniques: Inclusion-Exclusion, Applications of Inclusion exclusion principle, recurrence relations, solving recurrence relation. Relations: Relations and their properties, Binary Relations, Equivalence relations, Diagraphs, Matrix representation of relations and digraphs. Computer representation of relations and digraphs; Transitive Closures, Warshall's Algorithm, Problem solving on Warshall's Algorithm.	
UNIT 3: Ordered Sets and Graph Theory	**14 Hrs**
Partially Ordered Sets (Posets), External elements of partially ordered sets, Hasse diagram of partially ordered set, isomorphic ordered set, Lattices: Properties of Lattices, complemented Lattices. Graph theory: Introduction to graphs, Graph Terminology Weighted graphs, Representing Graphs, Connectivity of Graphs: Paths and Circuits, Eulerian and Hamiltonian Paths, Matrix representation of graphs. Graph Coloring and its applications.	
UNIT 4: Trees, Boolean Algebra, and Groups	**14 Hrs**

Trees: Rooted trees, Application of trees: Binary Search Trees, Decision Trees, Prefix Codes, Tree traversal, trees and sorting, spanning trees, minimal spanning trees. Finite Boolean algebra, Functions on Boolean algebra, Boolean functions as Boolean polynomials. Groups and applications: Subgroups, Semigroups, Monoids Isomorphism, Homomorphism.	

Textbooks

1. KENNETH H. ROSEN "Discrete Mathematics and Its Applications", 7th Edition, 2017 , Tata McGraw Hill

Reference Books

1. LIU, "Elements of Discrete Mathematics", 4th Edition, 2017, Tata McGraw Hill
2. SCHAUMS Outlines, "Discrete Mathematics", 3rd Edition, 2017, Tata McGraw Hill.
3. KOLMAN/REHMAN, "Discrete Mathematical Structures", 6th Edition, 2015, Pearson Education
4. NICODEMI "Discrete Mathematics",2002, CBS

COURSE OUTCOMES(CO):

CO1: Ability to Apply Logical Reasoning and Proof Techniques: Students will demonstrate proficiency in using propositional and predicate logic to construct valid arguments and proofs. They will apply methods such as direct proof, indirect proof, and mathematical induction to solve problems and analyze algorithms.

CO2: Competence in Counting and Probability Analysis: Students will be able to apply counting techniques such as permutations, combinations, and the Pigeonhole Principle to solve discrete probability problems. They will analyze recurrence relations and apply advanced counting principles like the Inclusion-Exclusion Principle.

CO3: Understanding and Application of Graph Theory and Relations: Students will acquire knowledge of graph theory, including graph representations, connectivity, paths, cycles, and graph coloring. They will understand properties of relations, matrix representations of relations and digraphs, and algorithms like Warshall's Algorithm for transitive closure.

CO4: Proficiency in Structural Analysis and Algebraic Concepts: Students will demonstrate proficiency in analyzing structures such as partially ordered sets (Posets), lattices, trees, and Boolean algebra. They will apply concepts of functions, groups, and monoids to solve problems in various applications, including decision trees, sorting, and Boolean functions.

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	2	2	1	2	3	2	2	3
2	2	2	1	1	1	-	-	1	2	1	1	2
3	3	3	2	2	2	2	1	2	3	2	2	3
4	2	2	2	2	2	2	-	2	2	2	2	2

COURSE TITLE: Computer Networks								
Course Code: MCA24104DCE						Examination Scheme	T	P
Total number of Lecture Hours:56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical's (P):	-	Tutorial (T):	-	Total Credits	4	

Course Objectives

- To gain a comprehensive understanding of the core principles of computer networking, including protocol design, protocol layering, algorithm design, and performance evaluation.
- To acquire detailed knowledge of the OSI model and TCP/IP protocol suite and understand the design issues and protocols used in the data link layer and MAC sublayer.
- To Understand the design issues of the network layer, including various routing algorithms and congestion control mechanisms.
- To learn about the protocols used in the transport and application layers, including their design and functionality.

Course Content	TEACHING HOURS
UNIT 1:	-14 Hrs
Introduction: Network, Uses of Networks, Types of Networks, Reference Models: TCP/IP Model, The OSI Model, and Comparison of the OSI and TCP/IP reference model. Architecture of Internet. **Physical Layer:** Guided transmission media, Wireless transmission media, Radio Transmission, Microware Transmission, Infrared Transmission and Light Transmission, Digital Modulation and Multiplexing, Switching. **Cellular Networks:** Cells, Handoff, Paging, Different Generation of Cellular Networks, GSM, Cable Networks, Communication Satellites and Policy at the Physical Layer.	
UNIT 2:	-14 Hrs
Data Link Layer: Design issues, Error Detection & Correction, Elementary Data Link Layer Protocols, Sliding window protocols and SONET **Medium Access Control Sub layer:** The Channel Allocation problem and Multiple Access Protocols, Ethernet. Multiple Access Protocols - ALOHA, CSMA,CSMA/CD, CSMA/CA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, Data link layer switching: Use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.	
UNIT 3:	-14 Hrs
Network Layer: Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Link State Routing, Path Vector Routing, Hierarchical Routing: Congestion control algorithms, IP addresses, CIDR, Subnetting, SuperNetting, IPv4, Packet Fragmentation, IPv6 Protocol, Transition from IPv4 to IPv6, ARP, RARP, OSPF , BGP and Traffic Prioritization.	
UNIT 4:	-14 Hrs
Transport Layer: Services provided to the upper layer's elements of transport protocol addressing connection establishment, Connection release, Error Control & Flow Control, Crash Recovery. The Internet Transport Protocols: UDP, Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control Algorithm, Socket Programming. **Application Layer:** Introduction, providing services, Applications layer paradigms: Client server model, HTTP, E-mail, WWW, TELNET, DNS.	

Textbooks

Andrew Tanenbaum, "Computer Networks", 6th Edition by Pearson, 2022

Reference Books

Behrouz A. Foruzan - Data communication and Networking, 6th edition, TMH, 2022

COURSE OUTCOMES (CO):

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

CO1: List the functionalities of different layers in both the OSI and TCP/IP reference models.

CO2: Analyze complex networking problems, including the concepts of internetworking, and the differences between connection-oriented and connection-less approaches.

CO3: Describe the principles of switching and routing algorithms used in computer networks.

CO4: Distinguish between TCP and UDP formats and procedures, understanding their respective uses and characteristics.

CO5: Identify, formulate, and analyze complex networking issues, applying principles and concepts learned throughout the course.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	3	3	3	2	2	1	-	2	-	-
2	3	2	2	2	-	2	1	1	-	-	-	-
3	3	2	2	2	1	1	1	1	-	1	-	2
4	3	2	1	1	1	0	1	1	-	1	1	1

COURSE TITLE: Computer Graphics									
Course Code: MCA24105DCE						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:

- Understand key concepts, graphic display devices, and 2D/3D transformations.
- Develop skills in line and circle drawing, clipping, filling, and hidden surface removal.
- Apply mathematical techniques like splines and Bezier methods for complex graphical models.
- Explore multimedia concepts, file formats, storage solutions, and introductory AR/VR technologies.

Course Content	TEACHING HOURS
UNIT 1:	**14Hrs**
Introduction to Computer Graphics, Applications of Computer Graphics, Graphic Display Devices: Refresh Cathode Ray Tubes, Raster-scan Displays, Random-Scan displays, Color CRT Monitors, Concept of Double Buffering, Lookup tables. 2-D Graphics: Cartesian and Homogeneous Coordinate Systems, Line drawing algorithms (Bressenham's and DDA), Circle and Ellipse Drawing Algorithms.	
UNIT 2:	**14Hrs**
2-Dimensional Transformations, Concepts of Window & Viewport, Window to Viewport Transformations, Normalization transformation (3L) Composite Transformations: General pivot point rotation, General fixed point scaling, reflection w.r.t line $y=x$, reflection w.r.t line $y=x$ (4L) Transformation between coordinate systems, affine transformations, Raster methods for transformations (3L)	
UNIT 3:	**14Hrs**
Filling techniques: Boundary and Flood-fill algorithms (2L) Clipping, Line Clipping Algorithms (Cohen-Sutherland Algorithm), 3-D Graphics, Projections: perspective and parallel projection transformations. (5L) 3-Dimensional Transformations, Hidden Surface Removal Techniques, Z-Buffer Algorithm, Back Face Detection (3L)	
UNIT 4:	**14Hrs**
Curves and Surfaces: Spline specification, Interpolated& Approximated Splines. spline representation, cubic spline interpolation methods, Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. (3LIntroduction to multimedia elements: Images (BMP, PCX), sound (WAV, MP3) Multimedia storage formats: CDs and DVDs). Introduction to virtual reality (VR) and augmented reality (AR) technologies.	
Textbooks	

1. Hearn and Baker, "Computer Graphics with OpenGL": 4th Edition (2022), Donald Hearn, M. Pauline Baker, Warren Carithers, Pearson
2. Ze-Nian Li and Mark S. Drew, "Fundamentals of Multimedia": 3rd Edition (2021), Springer.
3. W.M. Newman and Sproull, "Principles of Interactive Computer Graphics" McGraw-Hill Education; 3rd Edition, 2023.

Reference Books

1 **Steven Harrington, "Computer Graphics: A Programming Approach"** McGraw-Hill Education; 2nd Edition, 2021.
2 **Plastock and Kelley, "Schaum's Outline of Theory and Problems of Computer Graphics"** McGraw-Hill Education; 2nd Edition, 2022.
3 **David F. Rogers and J. Alan Adams, "Procedural Elements of Computer Graphics"** McGraw-Hill Education; 3rd Edition, 2021.
4 **David F. Rogers and J. Alan Adams, "Mathematical Elements of Computer Graphics"** McGraw-Hill Education; 3rd Edition, 2022.
5 **James D. Foley, Andries van Dam, et al., "Computer Graphics: Principles and Practice"** Pearson; 4th Edition, 2023.
6 **Sinha and Udai, "Computer Graphics"** Tata McGraw-Hill Education; 2nd Edition, 2022

COURSE OUTCOMES (CO):

CO1: Apply Graphics Principles: Understand and apply core concepts of computer graphics and transformations.
CO2: Implement Algorithms: Develop and execute line drawing, clipping, and filling algorithms.
CO3: Create Curves and Surfaces: Design and manipulate graphical models using spline and Bezier techniques.
CO4: Manage Multimedia: Integrate and manage multimedia elements and file formats.
CO5: Explore AR/VR: Apply basic concepts of augmented reality and virtual reality technologies.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Management Information Systems								
Course Code: MCA24106DCE						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

- To describe the role of information technology and decision support systems in business and record the current issues with those of the firm to solve business problems.
- To introduce the fundamental principles of computer-based information systems analysis and design and develop an understanding of the principles and techniques used.
- To enable students, understand the various knowledge representation methods and different expert system structures as strategic weapons to counter the threats to business and make business more competitive.
- 4. To enable the students to use information to assess the impact of the Internet and Internet technology on electronic commerce and electronic business and understand the specific threats and vulnerabilities of computer systems.

Course Content	TEACHING HOURS
UNIT 1: Basic Concepts of Information System	**-14 Hrs**
Role of data and information, Organization structures, Business Process, Systems Approach and introduction to Information Systems. Resources and components of Information System, integration and automation of business functions and developing business models. Role and advantages of Transaction Processing System, Management Information System, , Executive Support Systems and Strategic Information Systems.	
UNIT 2: Architecture & Design of IS and Decision Making Process	**-14 Hrs**
Architecture, development and maintenance of Information Systems, Centralized and Decentralized Information Systems, Factors of success and failure, value and risk of IS. Programmed and Non- Programmed decisions, Decision Support Systems, Models and approaches to DSS	
UNIT 3: Introduction to Enterprise Management technologies	**-14 Hrs**
Business Process Reengineering, Total Quality Management and Enterprise Management System viz. ERP, SCM, CRM and Ecommerce. An Overview of Enterprise an Overview of Enterprise; Integrated Management Information; Business Modeling; ERP for Small Business	
UNIT 4: : Security and Ethical Challenges	**-14 Hrs**

To be effective from year-2024

Ethical responsibilities of Business Professionals – Business, technology. Computer crime – Hacking, cyber theft, unauthorized use at work. Piracy – software and intellectual property. Privacy – Issues and the Internet Privacy. Challenges – working condition, individuals. Health and Social Issues, Ergonomics and cyber terrorism.	

Text Book:

Management Information Systems", W. S. Jawadekar, Tata McGraw Hill Edition, 6/e

Reference Books:

1. "Management Information Systems", Kenneth J Laudon, Jane P. Laudon, Pearson/PHI,10/e
2. Introduction to Information System", James A. O' Brien, Tata McGraw Hill, 12th Edtion.

COURSE OUTCOMES (CO):

CO1: Relate the basic concepts and technologies used in the field of management information systems;

CO2: Compare the processes of developing and implementing information systems.

CO3: Outline the role of the ethical, social, and security issues of information systems.

CO4: Translate the role of information systems in organizations, the strategic management processes, with the implications for the management

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Artificial Intelligence

Course Code: MCA24108DCE						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 Objective:

- To develop a solid understanding of the basic principles and history of artificial intelligence.
- Learn how to represent and organize knowledge for intelligent systems.
- Understand and apply reasoning methods for decision-making and problem-solving.
- To implement and apply algorithms to solve complex problems.

Course Content	TEACHING HOURS
UNIT 1: Introduction to Artificial Intelligence	**-14 Hrs**
Definition and history of artificial intelligence. AI applications and scope. Logic-based representation (Propositional logic, First-order logic), Knowledge-based systems and expert systems. Forward chaining, backward chaining. Agents: Intelligent agents, Agents and Environment, Structure of Agents Knowledge.	
UNIT 2: Fuzzy Logic	**- 14 Hrs**
Fuzzy logic and uncertainty. Fuzzification. Linguistic terms. Fuzzy sets. Hedges. Reasoning in Fuzzy Logic. Fuzzy set operations. Fuzzy vector matrix multiplication. Fuzzy Max-Min inferencing. FuzzyMax-Product inferencing. Multiple premise fuzzy inferencing. Mamdani Inference. Fuzzy multiple rule aggregation. Defuzzification. Applications of fuzzy logic.	
UNIT 3: Inductive Learning Algorithms	**-14 Hrs**
Inductive learning algorithms. Categories of inductive learning algorithms. Rule extraction with inductive learning algorithms, Decision trees, ID3 algorithm. AQ algorithm, SAFARI algorithm Applications of Inductive Learning Machine Learning: Supervised, Unsupervised and Reinforcement Learning.	
UNIT 4: Search Algorithms	**-14 Hrs**
Search Algorithms – Uninformed search strategies, Informed search strategies, Hill Climbing, Constraint satisfaction problems, Optimization techniques: Genetic algorithms, Simulated annealing, Ant colony optimization, Swarm Particle optimization	

Textbooks

1. "Artificial Intelligence: A Guide to Intelligent Systems" by Michael Negnevitsky, Latest Edition, 2020.
2. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, 4th Edition, 2020.
3. "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell, Latest Edition, 2019

Reference Books

1. "Artificial Intelligence" by Elaine Rich, Kevin Knight, and Shivashankar B. Nair, 4th Edition, 2021.
2. "Artificial Intelligence: Foundations of Computational Agents" by Michael Wooldridge, Ist Edition, 2021.
3. "Nature-Inspired Optimization Algorithms" by Saeid Aziznejad, Gholamreza Z. Naderpour, and Mohammad A. H. Sadeghi, Ist Edition, 2019.

To be effective from year-2024

COURSE OUTCOMES(CO):

CO1: Identify and discuss various applications of AI across different domains and their impacts.
CO2: Develop and implement knowledge-based systems and expert systems for decision-making and problem-solving.
CO3: Utilize constraint satisfaction problems and optimization techniques to tackle complex issues.
CO4: Implement and evaluate informed and uninformed search algorithms to solve problem-solving tasks.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Software Engineering								
Course Code: MCA24107CR						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

- **Understand Software Engineering Fundamentals**: Gain knowledge of the nature, goals, and challenges of software engineering and its historical context.
- **Apply Software Development Processes**: Learn and utilize various software development models, including Waterfall, Agile, and Spiral.
- **Measure Software Processes and Projects**: Analyze software processes using measures, metrics, and models like CMMI and COCOMO.
- **Master Requirements Engineering**: Develop skills in eliciting, analyzing, modeling, and validating both functional and non-functional requirements.
- **Design Engineering Proficiency**: Understand design principles, modularity, and patterns, and apply function-oriented and object-oriented design methodologies.
- **Achieve Competence in Software Testing and Reliability**: Understand core testing concepts and techniques, and explore software reliability and reengineering processes.

Course Content	TEACHING HOURS
UNIT 1: Fundamentals of Software Engineering	**14 Hrs**
Concept and Nature of Software: Concept and Nature of Software, Software Crisis, Software Engineering – Concept, Goals and Challenges, Software Engineering Approach. Software Development Process, Process Models - Waterfall Model, Evolutionary and Throwaway Prototyping Model, Incremental and Iterative Models, Spiral Model, Agile Process Model, Component based and Aspect Oriented development Software Process and Project Measurement: Measures, Metrics and Indicators, Size -Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model.	
UNIT 2: Requirements Engineering	**14 Hrs**
Introduction to Requirements Engineering - Why, What and Where. Requirements Types: functional and nonfunctional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques. Requirement Analysis and Modelling, Requirements prioritization, verification, and validation.	
UNIT 3: Design Engineering	**14 Hrs**
Basics of Design Engineering - Abstraction, Architecture, Patterns, Separation of concerns, Modularity, Functional Independence, refinement, Refactoring. Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology. Object-Oriented Design - Design Concepts, Design Methodology, Object-oriented analysis and design modeling using Unified Modeling Language (UML), Dynamic & Functional Modeling, Design Verification.	

UNIT 4: Software Testing and Reliability	14 Hrs
Software Testing – Concepts, Terminology, Testing & Debugging, Adequacy Criteria, Static vs. Dynamic Testing, Black Box vs. White Box Testing, Structural testing and its techniques, Functional Testing and its techniques, Mutation testing, Random Testing, Non-Functional Testing like Reliability, Usability, Performance and Security Testing. Introduction to Software Reliability: Basic Concepts, Correctness Vs Reliability, Software Reliability metrics, Operational Profile, Reliability Estimation and Predication, Reliability and Testing. Concept of Software reengineering, reverse engineering and change management.	

Textbooks

1. Shari Lawrence Pfleeger and Joanne M. Atlee - "Software Engineering: Theory and Practice," 4th Edition, Pearson, 2010.

Reference Books

1. Ian Sommerville - "Software Engineering," 10th Edition, Pearson, 2015.
2. Pankaj Jalote - "An Integrated Approach to Software Engineering," 3rd Edition, Narosa Publishing House, 2005.
3. Hans Van Vliet - "Software Engineering: Principles and Practice," 4th Edition, Wiley, 2016.
4. James F. Peters - "Software Engineering: An Engineering Approach," 1st Edition, Wiley & Sons, 2000.
5. Roger Pressman - "Software Engineering: A Practitioner's Approach," 8th Edition, McGraw-Hill Publications, 2014.

COURSE OUTCOMES (CO):

CO1: Students will explain the nature of software, the software crisis, and the goals and challenges of software engineering.

CO2: Students will implement appropriate software development models such as Waterfall, Agile, and Spiral based on project needs.

CO3: Students will assess software processes using metrics and models like CMMI and COCOMO.

CO4: Students will perform requirement elicitation, analysis, modeling, prioritization, verification, and validation.

CO5: Students will apply design principles and object-oriented design methodologies using UML.

CO6: Students will execute various testing techniques and evaluate non-functional requirements like reliability and performance.

LEVEL OF CO-PO MAPPING TABLE

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

To be effective from year-2024

COURSE TITLE: Block Chain Technologies								
Course Code: MCA24109DCE						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:

- Develop a deep understanding of the fundamental principles of blockchain technology, including distributed ledger technology (DLT), cryptographic methods, and consensus mechanisms, and apply this knowledge to evaluate different blockchain architectures.
- Analyze and compare various consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), and other emerging methods, understanding their impact on blockchain security, efficiency, and scalability.
- Demonstrate proficiency in blockchain development, including the creation and deployment of smart contracts using Solidity and other blockchain programming languages, and the development of decentralized applications (DApps) on platforms like Ethereum and Hyperledger.
- Assess the security challenges associated with blockchain and cryptocurrencies, including potential threats, vulnerabilities, and the implementation of secure coding practices to mitigate risks in blockchain applications.
- Investigate advanced applications of blockchain technology in fields such as IoT, AI, and healthcare, and evaluate the potential of blockchain to solve real-world problems in these domains.
- Critically analyze the future trends and emerging technologies in the blockchain ecosystem, such as quantum-resistant blockchains, decentralized identity solutions, and cross-chain interoperability, to understand their potential impact on industry and society.

Course Content	TEACHING HOURS
Unit 1: Introduction to Blockchain Technology	**14 Hrs**
Introduction to Blockchain - Definition, History, and Evolution. Basic Concepts - Distributed Ledger Technology (DLT), Cryptography, and Consensus Mechanisms. Types of Blockchains - Public, Private, Consortium, and Hybrid Block chains. Blockchain Structure - Blocks, Chains, Nodes, and Transactions. Cryptographic Foundations - Hash Functions, Digital Signatures, Public and Private Keys. Consensus Algorithms - Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS. Smart Contracts - Definition, Creation, Execution, and Security Issues. Overview of Major Blockchain Platforms - Bitcoin, Ethereum, Hyperledger .	
Unit 2: Blockchain and Cryptocurrencies	**14 Hrs**
Blockchain Networks - Nodes, Peer-to-Peer Networks, and Distributed Consensus. Security in Blockchain - Threats, Attacks, and Countermeasures. Blockchain Use Cases - Financial Services, Supply Chain, Healthcare. Introduction to Cryptocurrencies - Bitcoin and Altcoins. Bitcoin Architecture - Blockchain, Mining, Wallets, and Transactions. Ethereum and Smart Contracts - Solidity, DApps, and Gas. Cryptocurrency Wallets - Types, Security, and Key Management.	
Unit 3: Blockchain Development and Implementation	**14 Hrs**
Introduction to Blockchain Development - Tools, Platforms, and IDEs. Blockchain Development Languages - Solidity, Vyper, Go, and JavaScript. Building Smart Contracts - Basics, Writing, and Deploying. Developing DApps - Frontend, Backend, and Smart Contract Integration. Ethereum Development Environment - Truffle, Ganache, Remix. Hyperledger Fabric - Architecture, Components, and	

Development. Testing Blockchain Applications - Unit Tests, Integration Tests .	
Unit 4: Advanced Topics and Future Directions in Blockchain	**14 Hrs**
Blockchain in IoT - Use Cases, Challenges, and Solutions. Blockchain and Big Data - Integration, Analytics, and Use Cases. Blockchain in AI - Synergies, Applications, and Challenges. Blockchain and Cloud Computing - Decentralized Cloud Solutions. Green and Sustainable Blockchain Technologies. Quantum Computing and its Impact on Blockchain . Future Directions - Web 3.0, Decentralized Identity, and Tokenization of Assets.	

Textbooks:

1. "Blockchain Technology: Concepts and Applications" by Kumar Saurabh and Ashutosh Saxena, McGraw-Hill Education (2020).
2. "Cryptocurrency and Blockchain Technology" by Shaik Nasrullah and M. Balamurugan, Pearson (2021).
3. "Blockchain and Cryptocurrency" by B. B. Gupta and Hemraj Saini, PHI Learning (2020).

Reference Books:

1. "Cryptography and Blockchain Technology" by Atul Kahate, McGraw-Hill Education (2018).
2. "Blockchain: Principles and Applications" by Umesh Kumar Singh and Kavita Rani, Pearson (2020).
3. "Blockchain Technology and Applications" by M. S. Kiruthika and B. Prabu, PHI Learning (2021).

COURSE OUTCOMES (CO):

CO1: Students will be able to explain the core concepts of blockchain technology, including distributed ledgers, cryptographic principles, and consensus mechanisms, and demonstrate their application in real-world blockchain architectures.

CO2: Students will be capable of comparing and contrasting various blockchain consensus algorithms, assessing their impact on security, performance, and scalability, and determining the appropriate algorithm for specific use cases.

CO3: Students will acquire practical skills in blockchain development, including writing, testing, and deploying smart contracts using platforms like Ethereum and Hyperledger, and integrating these contracts into decentralized applications (DApps).

CO4: Students will be able to identify and analyze security threats and vulnerabilities in blockchain systems and cryptocurrencies, and implement secure coding practices and strategies to protect blockchain-based applications.

CO5: Students will critically evaluate the potential of blockchain technology in advanced applications, such as IoT, AI, & healthcare, and propose innovative solutions using blockchain to address challenges in these areas.

CO6: Students will demonstrate an understanding of emerging trends in the blockchain ecosystem, such as quantum-resistant blockchains, decentralized identity solutions, and cross-chain interoperability, and predict their potential impact on the future of the technology and its applications.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Computer Architecture & ALP								
Course Code: MCA24110DCE						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

- Describe the architecture of 8086 microprocessor
- Differentiate between various addressing modes of 8086 microprocessor
- Illustrate the instruction format of 8086 microprocessor
- Explain various types of instructions available in 8086 microprocessor
- Use emu8086 to write basic 8086 assembly programs
- Develop advanced 8086 assembly programs using procedures and INT 21H services

Course Content	TEACHING HOURS
UNIT 1: 8086 Architecture	**14 Hrs**
Basic features of 8086 Microprocessor, 8086 Microprocessor Architecture (BIU, EU, Instruction Queue). 8086 Programming model (General Purpose Registers, Segment Registers, Pointer & Index Registers, Flag & Other Registers). Segmentation in 8086. 8086 Pin-out diagram, 8086 Operating modes (Minimum and Maximum Mode), 8086 Addressing modes.	
UNIT 2: Instruction Set	**14 Hrs**
8086 Instruction Format 8086 Instruction Set, Data-transfer Instructions, Arithmetic Instructions, Logical/Bit-manipulation Instructions. Branching instructions, Looping instructions. Shift instructions, Rotate instructions, String instructions, Processor control instructions.	
UNIT 3: Assembly language	**14Hrs**
Introduction, Instruction Statement and Assembler Directives, TASM using emu8086. Data Definition Directives, Named Constant Directives, Simplified Segment Directives. TASM Memory Models Writing basic assembly programs in emu8086	
UNIT 4: Advanced assembly concepts	**14 Hrs**
Stack: Defining a stack, PUSH and POP instructions. Procedures: Defining and Calling a procedure. CALL and RET instructions. Passing parameters to procedures (via registers and Stack). Macros and other Assembler directives. INT 21H Keyboard Services, INT 21H Display Services, INT 21H File Services. Writing advanced assembly programs using procedures and INT 21H.	

Textbooks:

M.T. Savalia. 8086 Programming and Advanced Processor Architecture. Wiley India, 2012

Reference Books:

1. T.P. Skinner. An Introduction to 8086/8088 Assembly Language Programming. John Wiley, 1985
2. W. A. Triebel, A. Singh. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware, And Applications. Pearson Education, 2007.
3. B. B. Brey, The Intel Microprocessors: Architecture, Programming and Interfacing, Merrill, 2nd Edition, 1991.

COURSE OUTCOMES (CO):

CO1: Students will be able to describe the architecture of 8086 microprocessor
CO2: Students will be able to differentiate between various addressing modes of 8086 microprocessor
CO3: Students will be able to illustrate the instruction format of 8086 microprocessor
CO4: Students will be able to explain various types of instructions available in 8086 microprocessor
CO5: Students will be able to use emu8086 to write basic 8086 assembly programs
CO6: Students will be able to develop advanced 8086 assembly programs using procedures and INT 21H services

LEVEL OF CO-PO MAPPING TABLE

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

OE/GE

To be effective from year-2024

COURSE TITLE: Digital and Technological Solutions						
Course Code: MCA24001OE				Examination Scheme	T	P
Total number of Lecture Hours:28 Total number of Practical Hours:-				External	40	-
				Internal	10	-
Lecture (L):	2	Practicals(P):	-	Tutorial (T):	Total Credits	2

Course Objectives

- To gain familiarity with digital paradigms;
- To sensitize about role & significance of digital technology;
- To provide know how of communications & networks;
- To bring awareness about the e-governance and Digital India initiatives;
- To provide a flavor of emerging technologies - Cloud, Big Data, AI, 3D printing.

Course Content	TEACHING HOURS
UNIT 1: Digital Systems: Evolution, Applications, and Core Concepts	**-14 Hrs**
Introduction & Evolution of Digital Systems. Role & Significance of Digital Technology. Information & Communication Technology & Tools. Computer System & its working. Software and its types. Operating Systems: Types and Functions. Problem Solving: Algorithms and Flowcharts. Communication Systems: Principles, Model & Transmission Media. Computer Networks & Internet: Concepts & Applications, WWW, Web Browsers, Search Engines, Messaging, Email, Social Networking. Computer Based Information System: Significance & Types. E-commerce & Digital Marketing: Basic Concepts, Benefits & Challenges.	
UNIT 2: Digital Empowerment: e-Governance, Financial Tools, Cybersecurity, and Emerging Technologies	**- 14 Hrs**
Digital India & e-Governance: Initiatives, Infrastructure, Services and Empowerment. Digital Financial Tools: Unified Payment Interface, Aadhar Enabled Payment System, USSD, Credit / Debit Cards, e-Wallets, Internet Banking, NEFT/RTGS and IMPS, Online Bill Payments and PoS. Cyber Security: Threats, Significance, Challenges, Precautions, Safety Measures, & Tools, legal and ethical perspectives. Emerging Technologies & their applications: Overview of Cloud Computing, Big Data, Internet of Things, Virtual Reality, Blockchain & Cryptocurrency, Robotics, Machine Learning & Artificial Intelligence, 3-D Printing. Digital Signatures.	

Textbooks

1. F Masoodi, Digital and Technological Solutions, Ist Edition, BPB, 2024

COURSE OUTCOMES (CO):

CO1: Knowledge about digital paradigm;
CO2: Realization of importance of digital technology, digital financial tools, e-commerce;
CO3: Know-how of communication and networks;
CO4: Familiarity with the e-governance and Digital India initiatives;
CO5: An understanding of use & applications of digital technology;
CO6: Basic knowledge of machine learning and big data.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	3	3	3	2	2	1	-	2	-	-
2	1	2	2	2	-	2	1	1	-	-	-	-
3	1	2	2	2	1	1	1	1	-	1	-	2
4	2	2	1	1	1	0	1	1	-	1	1	1

COURSE TITLE: Data Processing using Spreadsheets							
Course Code: MCA24001GE					Examination Scheme	T	P
Total number of Lecture Hours: 28 Total number of Practical Hours: -					External	40	-
					Internal	10	-
Lecture (L):	2	Practical (P):	-	Tutorial (T):	Total Credits	2	

Course Objectives

- Proficiency in Spreadsheet Software: Understanding data management and organization in Excel.
- Advanced Data Analysis: Perform complex analyses with advanced formulas, functions, and PivotTables.
- Data Visualization and Automation: Create customized charts and automate tasks using macros.

Course Content	TEACHING HOURS
UNIT 1: Data Collection, Cleaning, and Transformation Using Spreadsheets	**14Hrs**
Introduction to Spreadsheets for Data Processing: Overview of spreadsheet software (e.g., Microsoft Excel, Google Sheets) for data handling. **Data Entry and Validation:** Accurate data entry techniques, using data validation tools to prevent errors. **Data Cleaning Techniques:** Using spreadsheet functions to identify and correct errors, handle missing data, and ensure consistency (e.g., TRIM, CLEAN, and FIND/REPLACE). **Data Transformation:** Sorting, filtering, and grouping data; applying formulas to transform raw data into useful formats (e.g., CONCATENATE, TEXT functions). **Data Integration:** Combining data from multiple sheets or sources using functions like VLOOKUP, HLOOKUP, and INDEX-MATCH.	
UNIT 2: Data Analysis, Visualization, and Reporting Using Spreadsheets	**14Hrs**
Descriptive Statistics in Spreadsheets: Using built-in functions for calculating mean, median, mode, variance, and standard deviation. **Exploratory Data Analysis (EDA):** Creating PivotTables and using conditional formatting to identify patterns, trends, and anomalies in data. **Data Visualization:** Creating charts and graphs (e.g., bar, line, pie, scatter) to visually represent data; using dynamic charts for interactive visualization **Creating Reports:** Compiling data insights into comprehensive reports with charts, tables, and narrative text using spreadsheets. **Advanced Spreadsheet Tools:** Introduction to automation using macros, collaborative features, and sharing options within spreadsheet applications.	

Textbooks

1. "Excel 2021 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach, Wiley, 1st edition (2021).
2. "Data Analytics Using Excel" by Seema Acharya and Subhashini Chellappan, McGraw Hill Education, 1st edition (2021).
3. Excel Data Analysis for Dummies" by Ankur Sharma, Wiley India, 1st edition (2022).
4. "Learn Excel 2019 Essentials" by Ritu Arora; BPB Publications, 1st Edition, 2019.

Reference Books

1. Microsoft Excel 2021 Data Analysis and Business Modeling" by Wayne Winston, Microsoft Press, 1st edition (2022). ISBN: 9780137613663
2. Advanced Excel 2019" by Lokesh Lalwani; BPB Publications, 1st Edition, 2019.

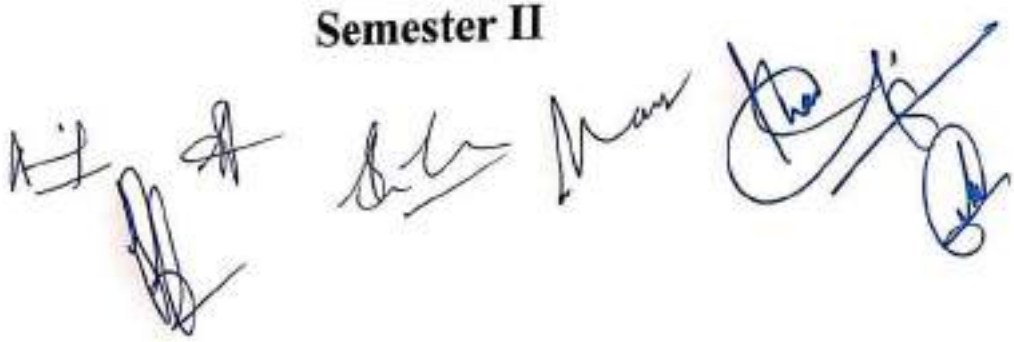
COURSE OUTCOMES (CO):

CO1: Efficiently organize and manage data within spreadsheets.
CO2: Utilize complex formulas, functions, and PivotTables for in-depth data analysis.
CO3: Create and format charts to effectively present data insights.
CO4: Automate repetitive tasks using macros to enhance productivity

LEVEL OF CO-PO MAPPING TABLE

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

Semester II



To be effective from year-2024

COURSE TITLE: Data Structures Using C++								
Course Code: MCA24201CR						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):	0	Total Credits	6	

Course Objectives

- Understand and implement linear data structures such as arrays and linked lists, including operations like insertion, deletion, and searching.
- Master stack and queue operations, including their representations in memory and implementations using arrays and linked lists.
- Comprehend tree structures including binary trees, binary search trees, AVL trees, and B-trees, along with their traversal techniques and applications.
- Learn graph terminology, representations, traversal techniques, and practical applications in computer science.
- Explore advanced data structures such as threaded binary trees, M-way search trees, and various types of heaps.
- Study file organization techniques including sequential, relative, and indexed sequential file organizations, as well as multiple key file organizations like inverted files and multi-list organizations.

Course Content	TEACHING HOURS
Unit I: Linear Data Structures	**14 Hrs.**
Data types/objects/structures, Data structures and its types, Representation and implementation. Linear Data Structures: Array representation, operations, applications and limitations of linear arrays, Searching Techniques- Linear Search, Binary Search Sorting Techniques- Selection, Insertion sort, Bubble sort, Quick Sort, Merge Sort Two dimensional arrays, matrices, common operations of matrices, special matrices, Array representation of Sparse matrices. Linked Lists: Representation, Types and operations on Linked List.	
Unit II: Stack and Queues	**14 Hrs.**
Stack- Representation of stack in memory, Operations on Stacks, Implementation of Stack using arrays and linked list, Multiple Stacks: Representing two stacks and more than two stacks, Applications of stacks: Parenthesis Checker, Infix to postfix procedure, evaluating expressions in postfix notation, Implementation of recursion using stack. Queues- Representation of Queue in Memory, Operations on Queue, Implementation of Queue using arrays and linked list, Circular Queue and its operations, Representation and implementation, Multiple Queues, Deque, Priority Queue, Heap Representation of a Priority Queue, Applications of Queues.	
Unit III: Tree and Graph Data Structures	**14 Hrs.**

Trees, Definitions, terminologies and properties, Binary tree representation, traversals and applications, Threaded binary trees, Binary Search Trees, AVL Trees, M-way Search Trees, B-trees, B+ trees, Graphs, Terminology, Graph representations, Traversal Techniques, Operations on Graphs, Applications of Graphs	
Unit IV: Advanced Data Structures and Algorithms	**14 Hrs.**
Minimum spanning trees, Shortest Path Algorithms in Graphs, Eulerian Tour, Hamiltonian Tour Hashing: Direct Address Tables, Hash Table, Different Hash functions, resolving collisions, rehashing, Heap Structures, Binomial Heaps, Leftist Heaps. File Organizations: Sequential File Organization, Relative File Organization, Indexed Sequential File Organization, Multiple Key File Organizations: Inverted File and Multi-List Organizations	

Textbooks

1. Langsam, Augenstein, Tenenbaum, "Data Structures Using C and C++", 2nd Edition, 2015

Reference Books

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson Freed, "Fundamentals of Data Structures In C", 2nd Edition, 2018
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 3rd Edition, 2007.
3. Aho Alfred V., Hopcroft John E., Ullman Jeffrey D, "Data Structures and Algorithms", 2017
4. R. S. Salaria, "Data Structures and Algorithms Using C++", 2018
5. Varsha H Patil, "Data Structures using C++", 2012
6. E.Balagurusamy, "Object Oriented Programming with C++", 8th Edition, 2020

Lab Manual

Week 1

- Write a program in C++ to insert, delete, and update the contents of an array.
- Write a program in C++ to search an element in an array.
- Write a program in C++ to perform various operations on matrices.
- Write a program in C++ to implement different string manipulation operations?
- Write a program to search an element in array using Binary Search.
- Write a program to implement Selection sort
- Write a program to implement bubble sort

Week 2

- Write a program to implement insertion sort
- Write a program to implement quick sort
- Write a program to implement merge sort
- Write a program to add two sparse matrices?
- Write a program to multiply two sparse matrices?

Week 3

- Write a program to implement singly linked list?
- Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a linkedlist.
- Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a linkedlist.
- Write a program in C++ to reverse a linked list by changing the link in the nodes?

Week 4

- Write a program to add two polynomials represented as linked list?
- Write a program in C++ to multiply two polynomials represented as linked lists?
- Write a program in C++ to implement a doubly linked list?
- Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linkedlist.
- Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, after a certain count of nodes in a doubly linkedlist.

Week 5

- Write a program to implement different operations of a circular linked list.
- Write a program to implement various operations on an array based stack?
- Write a program to implement various operations on a stack represented using linked list.

Week 6

- Write a program to demonstrate the use of stack in checking whether the arithmetic expression is properly parenthesized?
- Write a program to demonstrate the use of stack in converting an arithmetic expression from infix to postfix?
- Write a program to demonstrate the use of stack in evaluating an arithmetic expression in postfix notation?

Week 7

- Write a program to demonstrate the use of stack in implementing quicksort algorithm to sort an array of integers in ascending order.
- Write a program to demonstrate the implementation of various operations on a linear queue represented using a linear array
- Write a program to demonstrate the implementation of various operations on a Circular queue represented using a linear array.

Week 8

- Write a program to demonstrate the implementation of various operations on a queue represented using a linked list?
- Write a program to demonstrate the use of multiple stacks?

Week 9

- Write a program to delete a node in a binary search tree?
- Write a program to implement the different operations of an AVL tree
- Write a program to implement the different operations of a threaded binary tree.
- Write a program to implement the different operations of a M-way search tree?

Week 10

- Write a program to implement the different operations of a B- tree?
- Write a program in C++ to implement the different operations of a B+tree.
- Write a program in C++ to implement the graph using different representations.

Week 11

- Write a C++ program to illustrate the traversal of a graph using Breadth FirstSearch.
- Write a C++ program to illustrate the traversal of a graph using Depth FirstSearch.
- Write a program in C++ to find the edges of a spanning tree using Prims Algorithm.
- Write a program in C++ to find the shortest path in a graph using Warshalls Algorithm.

Week 12

- Write a C++ program to in C++ to find the shortest path in a graph using Dijkstra's Algorithm.
- Write a C++ program in C++ to implement Euler Graphs?
- Write a program in C++ to implement Hamilton Graphs?

Week 13

- Write a program in C++ to implement Planner Graphs?
- Write a program to C++ to implement Kruskals Algorithm?

Week 14

- Write a C++ program to implement a simple hash table using linear probing to resolve collisions.
- Write a C++ program to create Max and Min heaps?

COURSE OUTCOMES (CO):

CO1: Students will be able to implement and manipulate linear data structures such as arrays, linked lists, and matrices, including operations like insertion, deletion, and traversal.

CO2: Students will demonstrate proficiency in implementing and applying advanced data structures such as stacks, queues, trees (binary trees, AVL trees), graphs, and various heaps (binomial heaps, leftist heaps) to solve complex problems.

CO3: Students will understand and apply different file organization techniques such as sequential, relative, and indexed sequential file organizations, and multiple key file organizations like inverted files and multi-list organizations.

CO4: Students will develop analytical and problem-solving skills by applying appropriate data structures and algorithms to solve practical problems related to data storage, retrieval, and manipulation in computer science applications.

To be effective from year-2024

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	3	3	3	2	2	2	2	2	2	2	2	-
3	2	2	2	1	-	-	-	2	2	-	-	-
4	3	3	2	-	2	1	-	2	2	2	-	-

COURSE TITLE: Data Science with Python

Course Code: MCA24202CR						**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):**	**0**	**Total Credits**	**6**	

Course Objectives

- Gain a comprehensive understanding of the fundamental concepts, evolution, and scope of data analytics, including big data and different types of analytics.
- Learn the fundamentals of Python programming, including data types, control flow, and essential packages for data analysis.
- Explore key elements of machine learning, including supervised and unsupervised learning, and apply techniques such as regression and classification.
- Understand and apply various classification methods, including logistic regression, K-NN, and SVM, along with model evaluation techniques.

Course Content	**TEACHING HOURS**
UNIT 1: Foundation of Data Analytics:	**14 Hrs.**
Introduction to Data Analytics , Evolution, Concept and Scopes Big Data, Metrics and Data classification, Data Reliability & Validity, Problem Solving with Analytics Different phases of Analytics in the business and Data science domain Types of Data Analytics - Descriptive Analytics, Predictive Analytics, Prescriptive Analytics, Applications of Data Analytics Text Analytics and Web Analytics, Skills for Business Analytics Concepts of Data Science, Basic Skills for Data Science	
UNIT 2: Fundamentals of python	**14 Hrs.**
Introduction to Python - Editors & Interactive Development Environments; Custom environment settings for Jupyter,Spyder, PyCharm. Basic data types -numeric, string, float, tuples, list, Python Dictionary, sets and their operations Control flow in python - (if-elif-else), loops (for, while) Inbuilt functions for data conversion, Writing user defined functions in Python Important packages – NumPy, SciPy, Scikit-learn, Pandas, Matplotlib, Seaborn, etc; Installing and loading packages in Python Reading and writing data from/to different formats Python Multi-threaded Programming Plotting in python, functions, list comprehensions, Database connectivity in python, Playing with Date Format.	
UNIT 3: Feature Engineering with Machine Learning	**14 Hrs.**
Introduction, Definitions and types of machine learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning. Reinforcement and Transfer Learning Basics of Regression, Classification, Clustering	

Logistic Regression with one variable and with multiple variables, Application to multi-class classification, The problem of Overfitting, Application of Regularization in Linear and Logistic Regression Regularization and Bias/Variance.	
UNIT 4: Classification and Model Evaluation Techniques	**14 Hrs.**
Classification Using Logistic Regression, Logistic Regression vs. Linear Regression Classification using K-NN, Naive Bayes classifier, Decision Trees Linear Classification using Support Vector Machines Non Linear Classification using Support Vector Machines Cross validation types (train & test, bootstrapping, k-fold validation), Model Performance – Training, Validation and testing; Confusion matrices , Basic evaluation metrics, precision-recall, ROC curves.	

Textbooks:

1. **Jake VanderPlas**, *"Python Data Science Handbook"*, O'Reilly Media, 2016
2. **Joel Grus**, *"Data Science from Scratch"*, O'Reilly Media
3. **Madhusree Ghosh**, *"Data Science and Machine Learning"*, Springer

Reference Books:

Lab Manual

Week 1

- Install Python and set up IDEs like Jupyter Notebook or VS Code
- Write a "Hello, World!" program.
- Write a program to perform basic arithmetic operations: addition, subtraction, multiplication, and division.
- Write a program to print your name and age.

Week 2

- Write a program to create variables of different data types (int, float, complex, string) and print their values.
- Write a program to perform string operations: concatenation, slicing, and repetition.
- Write a program to demonstrate arithmetic, logical, and relational operations.

Week 3

- Write a program to create a list, perform slicing, and append elements to it.
- Write a program to demonstrate the use of tuple data type and its operations.
- Write a program to find the length, maximum and minimum value of a list.

Week 4

- Write a program to demonstrate the use of if, else, and elif statements.
- Write a program to print the first 10 natural numbers using a for loop.
- Write a program to print a pattern using nested loops (e.g., a pyramid).

Week 5

- Write a program to iterate over a string, list, and dictionary using loops.
- Write a program to demonstrate the use of while loops.
- Write a program to manipulate loops using pass, continue, break, and else.

Week 6

- Write a program to to define and call a function that adds two numbers.
- Write a program to demonstrate the use of lambda functions.
- Write a program with a function that takes a list as an argument and returns the sum of all its elements.

Week 7

- Write a program to create and import a custom module.
- Write a program to to use an external library (e.g., math or random).
- Write a program to organize code into a package.

Week 8

- Write a program to define a class and create objects.
- Write a program to demonstrate inheritance.
- Write a program to show polymorphism using method overriding.

Week 9

- Write a program to perform basic array operations with numpy arrays.
- Write a program to create and manipulate DataFrame objects using Pandas.
- Write a program to draw basic plots in Python program using Matplotlib.
- Write a program to perform a basic statistical analysis using SciPy.

Week 10

- Write a program to Count the frequency of occurrence of a word in a body of text is often needed during text processing..
- Write a program to compute weighted averages in Python either defining your own functions or using Numpy.
- Write a python program to calculate the mean, median, mode, variance.

Week 11

- Write a program to create a normal curve using python program.
- Write a python program for correlation with scatter plot
- Write a python program to compute correlation coefficient.

Week 12

- Write a program to demonstrate Regression analysis with residual plots on a given data set.
- Write a program to demonstrate the working of the decision tree-based ID3 algorithm.

Week 13

- Write a program to implement the Naïve Bayesian classifier for a sample training data set.
- Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set.

COURSE OUTCOMES (CO):

CO1: Understand and describe the evolution, concepts, and scope of data analytics.
CO2: Identify and classify different types of data analytics and their applications in various domains.
CO3: Utilize Python programming language for data analysis, including data handling and visualization.
CO4: Implement machine learning techniques such as regression, classification, and clustering for data analysis.
CO5: Evaluate the performance of different machine learning models using appropriate metrics.
CO6: Apply advanced classification techniques and understand their applications in real-world scenarios.

CO	PO											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	1	1	1	2	-	2	1	1	1	1	1
2	1	2	1	1	2	1	2	1	2	2	2	2
3	2	1	3	2	3	2	3	1	2	2	2	2
4	2	1	2	2	2	2	1	1	2	2	2	2
5	1	1	3	3	2	2	2	1	2	2	2	2
6	2	1	3	3	2	2	2	1	2	2	2	2

COURSE TITLE: Operating Systems

Course Code: MCA24203CR						**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):**	**0**	**Total Credits**	**4**	

Course Objectives

- Understand Fundamental Concepts of Operating Systems
- Develop Skills in Process Management and Synchronization
- Explore Distributed Operating Systems
- Gain Expertise in Deadlocks Management
- Explore Real Time Operating System
- Acquire Skills in Real-Time Task Scheduling

Course Content	**TEACHING HOURS**
UNIT 1: Introduction and Scheduling	**14 Hrs.**
Operating System Overview, Types of Operating Systems; Basic Operating System: Processes, Scheduling criteria, Scheduling Algorithms. Introduction to Distributed Operating System, Processor allocation and scheduling in distributed systems - System Models, Load balancing and sharing approach, fault tolerance. Introduction to Real Time Operating System, Basic OS Principles and Structures review; Real-Time Systems: Basic Model, Characteristics, Hard vs. Soft. Classification of Real-Time Scheduling Algorithms; Common Approaches; Clock Driven; Priority Driven: Earliest Deadline First, Rate Monotonic, Deadline Monotonic	
UNIT 2: Inter-Process Communication and Synchronization	**14 Hrs.**
Interprocess Communication and Synchronization, Classical problems, Critical section, Semaphores, Monitors. Synchronization in Distributed Systems; Clock Synchronization and related algorithms, Logical Clocks. Mutual Exclusion: Centralized & Distributed (Contention & Token) Algorithms. Election Algorithms: Bully Algorithm, Invitation Algorithm. Client Server model; Remote procedure call and implementation issues. Synchronization in RTOS; Resource Sharing among Real-Time Tasks – Contention and Control; Priority Inversion; Priority Inheritance Protocol; Highest Locker Protocol; Priority Ceiling Protocol	
UNIT 3: Memory Management	**14 Hrs.**
Memory Management: Address Spaces, Virtual Memory. Page Replacement Algorithms, Design and Implementation Issues for Paging Systems, Segmentation. General architecture of Distributed Shared Memory systems; Design and implementation issues of DSM; granularity - Structure of shared memory space, consistency models, replacement strategy, thrashing. Memory Technologies in RTOS; Different Classes of Memory, Memory Access and Layout Issues, Hierarchical Memory Organization [5 Lectures]	
UNIT 4: Deadlocks	**14 Hrs.**

Deadlocks characterization, Methods for handling deadlocks; Deadlock Prevention, Avoidance, Detection, Recovery.
Deadlocks in distributed OS; Deadlock Modeling, Handling Deadlocks in Distributed Systems, Deadlock Avoidance, Deadlock Prevention, Deadlock Detection; Centralized Approach for Deadlock Detection, Fully Distributed Approaches for Deadlock Detection, WFG-Based Distributed Algorithm for Deadlock Detection, Recovery from Deadlock, Issues in Recovery from Deadlock.
Deadlocks in RTOS

Textbooks:

1. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, "Operating System Principles", John Wiley.
2. Pradeep K. Sinha , "Distributed Operating Systems : Concepts and Design", PHI
3. Rajib Mall, Real-Time Systems: Theory and Practice (Second Edition), Pearson Education.

Reference Books:

1. Andrew S. Tanenbaum, "Modern Operating Systems", PHI. Andrew. S. Tanenbaum,"Distributed Operating System", PHI.
2. Andrew S. Tanenbaum, Modern Operating Systems (Third Edition), Pearson Education.
3. David E. Simon, An Embedded Software Primer, Pearson Education.
4. Laplante, P., Real-Time Systems Design and Analysis (Third Edition), IEEE/Wiley Interscience.
5. Jane W.S. Liu, Real-Time Systems (Sixth Edition), Pearson Education.
6. Raj Kamal, Embedded Systems: Architecture, Programming and Design (Third Edition), Tata McGraw-Hill Education

COURSE OUTCOMES (CO):

CO1: Students will understand the fundamental concepts and functions of an operating system.
CO2: Students will develop skills in process management and CPU scheduling techniques.
CO3: Students will acquire comprehensive knowledge of memory management methods and their practical applications.
CO4: Students will achieve proficiency in the principles and design of distributed systems.
CO5: Students will gain expertise in identifying, preventing, and resolving deadlocks.
CO6: Students will acquire expertise in real-time systems.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Cryptography and Network Security

Course Code: MCA24204DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- To gain a comprehensive understanding of the OSI Security Architecture and fundamental security concepts.
- To develop proficiency in cryptographic techniques and number theory.
- To master key management and authentication protocols.
- To apply cryptographic methods to network security and intrusion detection.

Course Content	TEACHING HOURS
UNIT 1: Security Fundamentals and Number Theory	**14 Hrs.**
Part 1: Information and Network Security Concepts. The OSI Security Architecture: -Security Attacks: Passive Attacks, Active Attacks, Threats and Vulnerabilities, Malware. OWASP top ten vulnerabilities. -Security Services: CIA, AAA, X.800 -Security Mechanisms: Specific security mechanisms and Pervasive security mechanism. **Part 2:** Introduction to Number Theory: Divisibility and the Division Algorithm, The Euclidean Algorithm, Modular Arithmetic, Prime and relatively prime Numbers, Fermat's and Euler's Theorems, Euler's Totient function, Testing for Primality, The Chinese Remainder Theorem. **Part 3:** Introduction to Cryptology, Classical Encryption Techniques, Substitution Techniques: Monoalphabetic Ciphers and Polyalphabetic Ciphers. Transposition Techniques, One Time Pad	
UNIT 2: Modern Cryptographic Techniques and Algorithms	**14 Hrs.**
Part 1: Block Ciphers: Data Encryption Standard: DES Structure, DES Example, the Strength of DES, Advanced Encryption Standard: AES Structure, AES Transformation Functions, AES Key Expansion, An AES Example. Block Cipher Modes of Operation: Electronic CodeBook, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode. **Part 2:** Random Bit Generation and Stream Ciphers: Principles of Pseudorandom Number Generation, Pseudorandom Number Generators, Pseudorandom Number Generation Using a Block Cipher, Stream Ciphers, RC4, Stream Ciphers Using Feedback Shift Registers. **Part 3:** ASYMMETRIC CIPHERS: Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm, Diffie–Hellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.	
UNIT 3: CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS	**14 Hrs.**

Part 1: Cryptographic Hash Functions SHA-1, SHA-3, Applications of Cryptographic Hash Functions, Two Simple Hash Functions. **Part 2:** Message Authentication Codes, MACs Based on Hash Functions: HMAC; MACs Based on Block Ciphers: DAA and CMAC, Authenticated Encryption: CCM and GCM, Key Wrapping, Pseudorandom Number Generation Using Hash Functions and MACs. **Part 3:** Digital Signatures: RSA Digital signature scheme, ElGamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm. Key management and distribution.	
UNIT 4: NETWORK AND INTERNET SECURITY	**14 Hrs.**
Part 1: Secure Shell /Transport-Level Security: Web Security Considerations, Transport Layer Security, HTTPS, Secure Shell (SSH). **Part 2:** Electronic Mail Security, IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload. **Part 3:** Network Endpoint Security: Firewalls, Intrusion Detection Systems.	

Textbooks

1. William, Stalling, Cryptography and Network Security, 8/E." Prentice Hall. (2023).
2. Forouzan, Behrouz A., and Debdeep Mukhopadhyay. Cryptography and network security (Sie). McGraw-Hill Education, 2011.

Reference Books

1. Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.
2. Introduction to Modern Cryptography (Chapman & Hall/CRC Cryptography and Network Security Series) Jonathan Katz , Yehuda Lindell

COURSE OUTCOMES (CO):

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

CO1: Explain the fundamental concepts of cryptography, including symmetric and asymmetric encryption, hashing, digital signatures, and key management.
CO2: Understand the historical development and relevance of cryptographic techniques in modern security protocols.
CO3: Apply various cryptographic algorithms, such as AES, RSA, ECC, and SHA, to secure data and communications.
CO4: Analyze and critically evaluate the strengths and weaknesses of different cryptographic protocols, such as SSL/TLS, IPsec, and PGP.
CO5: Design and implement network intrusion detection systems, integrating cryptographic solutions to protect against various security threats and attacks.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Digital Image Processing

Course Code: MCA24205DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Develop a thorough understanding of the fundamental concepts and theories in image processing, including pixel representation, color spaces, and digital image formation
- Equip students with the technical skills to apply various image processing techniques such as image transformations, filtering, enhancement, and segmentation using appropriate software tools.
- Enhance students' ability to analyze and interpret images by implementing feature extraction and pattern recognition methods, and applying these techniques to solve real-world problems.
- Foster the ability to integrate image processing techniques into broader applications, such as computer vision, medical imaging, and multimedia, through project-based learning and case studies.

Course Content	TEACHING HOURS
UNIT 1: Introduction.	**14 Hrs.**
Introduction Digital Image processing, Origins of DIP, Examples, Fundamental steps in DIP, Components of DIP. Fundamentals Elements of visual perception: brightness, contrast, hue, saturation, Mach-band effect; Light and the electromagnetic spectrum. Image formation and digitization concepts; Image Sensing and acquisition; Image samplingand quantization. Basic relationships between pixels: Neighbours of pixel adjacency connectivity, regions andboundaries, Distance measures.	
UNIT 2: Image Enhancement	**14 Hrs.**
Image enhancement in the spatial domain: Background; Point and arithmetic/ logic operations; Some basic grey level transformations; Histogram processing: Equalization,Matching. Mechanics of spatial filtering: Correlation, Convolution; Smoothing spatial filters: Averaging and Weighted-Averaging Filters, Gaussian Filter; Sharpening spatial filters: First and Second Derivatives, Laplacian, Unsharp Masking and High Boost Filtering. Image enhancement in the frequency domain: Background, Introduction to the Fourier transform and the frequency domain, Smoothing Frequency-Domain filters, Sharpening Frequency Domain filters.	
UNIT 3: Image Restoration and Morphological Processing.	**14 Hrs.**

Model of image degradation/restoration process: Noise models; Restoration by spatial filtering: Mean Filters, Order-Statistics Filters; Restoration by frequency domain filtering: Bandreject Filters, Bandpass Filters. Morphological Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, connected components, thinning, thickening, skeletons,pruning. Color Image Processing: Color Fundamentals, Color Models: RGB, CMY and CMYK, HIS,Conversion from RGB to HSI and vice versa	
UNIT 4: Edge Detection and Segmentation.	**14 Hrs.**
Edge detection: Basic Formulation: Detecting Points and Lines, Edge Models; Gradient and its Properties; Gradient Operators: Roberts, Prewitt, Sobel; Canny Edge Detector; Thresholding: Basic Global Thresholding, Basic Adaptive Thresholding. [6 Lectures] Region based segmentation: Basic Formulation, Region growing, Region splitting and Merging; Segmentation by morphological watersheds: Basic concepts, Dam construction, Watershed Algorithm.	

Textbooks:

1. Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing, Pearson, SecondEdition, 2004.
2. Anil K. Jain. Fundamentals of Digital Image Processing, Pearson 2002.

Reference Books:

1. Principles of Digital Image Processing by Wilhelm Burger.

COURSE OUTCOMES (CO):

CO1: The students will be able to understand the fundamental principles of image processing, including pixel representation and colour spaces.
CO2: Students will be able to apply image transformation techniques such as scaling, rotation, and translation.
CO3: The students will be able to implement and use various image filtering techniques for noise reduction and edge detection.
CO4: Students will be able to apply image enhancement methods to improve image quality, such as histogram equalization and contrast adjustment.
CO5: Students will be able to perform image segmentation using techniques like thresholding and region-based methods.
CO6: Students will be able to extract and analyse key features from images for pattern recognition and classification tasks.

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: Decision Support Systems

Course Code: MCA24206DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- **Understand Decision Support Systems (DSS):** Gain a comprehensive understanding of Decision Support Systems, including their importance in enhancing decision-making processes within organizations.
- **Explore Development Methodologies:** Analyse both traditional and alternative methodologies for DSS development, focusing on their applications, advantages, and limitations. Understand how to manage change effectively during the development and implementation phases.
- **Evaluate DSS Technologies and Tools:** Learn about the various technology levels, development platforms, and tools available for DSS. Develop skills in selecting appropriate tools based on specific needs and technological constraints.
- **Study DSS Components and Models:** Understand the core components and characteristics of DSS. Explore different modelling techniques, including static and dynamic models, and how they handle certainty, uncertainty, and risk. Learn to use influence diagrams and construct mathematical models for decision support.
- **Implement Enterprise DSS:** Examine how DSS supports communication, collaboration, and group decision-making within organizations. Explore the role of enterprise information systems and executive support systems in organizational decision-making and transformation.
- **Facilitate Knowledge Management:** Understand the importance of knowledge management initiatives and approaches. Explore how DSS can aid in organizational learning, knowledge management, and the implementation of effective knowledge management strategies.

Course Content	TEACHING HOURS
UNIT 1: Decision Making	**14 Hrs.**
DSS Development Introduction – Traditional and alternative development methodologies - Change Management – DSS Technology Levels and Tools – Development Platforms – Tool Selection..	
UNIT 2: Modeling and Analysis	**14 Hrs.**
Definition – Characteristics and capabilities of DSS – DSS components - Modeling and issues – Static and dynamic models – Certainty, Uncertainty and Risk – Influence Diagrams – Structure of Mathematical models.	
UNIT 3: DSS Development	**14 Hrs.**
Introduction – Traditional and alternative development methodologies - Change Management – DSS Technology Levels and Tools – Development Platforms – Tool Selection.	
UNIT 4: Enterprise DSS and Knowledge Management	**14 Hrs.**

Communication support – Collaboration support - Group support systems and technologies – GSS meeting process – Creativity and idea generation – Enterprise information systems – Evolution – Characteristics and capabilities of executive support systems – Organizational DSS - Organizational learning and transformation – Knowledge management initiatives – approaches – implementation.	

Textbooks

1. Efraim Turban, Jay E Aronson, Ting Peng Liang. Decision Support and Intelligent Systems, Prentice Hall of India, 7th Edition 2005.
2. Efraim Turban, Ramesh Sharda, Dursun Delen, Decision support and Business Intelligence systems, Pearson Education, 9th Edition, 2011.

Reference Books: -

1. Decision Support systems for business Intelligence 2nd edition by Vicki L Sauter Willey
2. Elain Rich and Kevin Knight, Artificial intelligence, TMH, 2006

COURSE OUTCOMES (CO):

CO1: Understand concept of managerial decision systems and outline its various phases.
CO2: Demonstrate DSS components and identify sources of data for business intelligence.
CO3: Categorize the methodologies involved in DSS development.
CO4: Analyze evolution of enterprise DSS and knowledge management initiatives.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Software Project Management

Course Code: MCA24207DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- To provide fundamental skills of software Project management emphasizing on issues & hurdles associated with delivering successful projects.
- Apply project management concepts through working in a group as team leader or active team member on an IT project.
- Utilize scheduling terminology, techniques, and tools to create accurate and feasible project timelines.
- Develop and use Bar Charts, Milestone Charts, and Gantt Charts for tracking project progress and communicating schedules.

Course Content	TEACHING HOURS
UNIT 1: Introduction to SPM	**14 Hrs.**
Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope Document, Project Management Cycle, SPM Objectives SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of Project Plan, Structure of a Software Project Management Plan Software Project Estimation, Estimation Methods, Estimation Models, Decision Process	
UNIT 2: Project Organization and Scheduling Project Elements	**14 Hrs.**
Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle Ways to Organize Personnel, Project Schedule, Scheduling Objectives, Building the Project Schedule, Scheduling Terminology and Techniques Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts	
UNIT 3: Project Monitoring and Control	**14 Hrs.**
Dimensions of Project Monitoring & Control, Earned Value Analysis Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI) Software Reviews, Types of Review: Inspections, Deskchecks, Walkthroughs, Code Reviews	
UNIT 4: Software Quality Assurance	**14 Hrs.**

Concept of Software Quality, Software Quality Attributes, Software Quality Metrics and Indicators, The SEI Capability Maturity Model (CMM) SQA Activities, Formal SQA Approaches: Proof of Correctness, Statistical Quality Assurance, Product versus process quality management, Introduction, types of contracts, stages in contract, placement, typical terms of a contract, contract management, acceptance.		

Textbooks:-

1. Software Project Management, Bob Hughes and Mike Cotterell, McGraw Hill

Reference Books: -

1. Software Project Management A Unified Framework, Walker Royce, Addison-Wesley
2. A practitioner's Guide to Software Engineering, Roger Pressman, Tata McGraw Hill 2014 8th edition.
3. Basics of Software Project Management, NIIT, Prentice-Hall India, Latest Edition

COURSE OUTCOMES (CO):

CO1: Define the principles of project management for developing software.
CO2: Explain various project management scheduling techniques.
CO3: Apply different techniques of project monitoring, control and review.
CO4: Classify various project management tools and estimate the risks involved in project activities.
CO5: Assess issues related to project quality and staffing.
CO6: Discuss the effect of project management practices in an organization

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Machine Learning

Course Code: MCA24208DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives:

- Equip students with a deep understanding of core machine learning techniques, including clustering, classification, dimensionality reduction, and neural networks, with a focus on both theoretical concepts and practical implementations.
- Enable students to apply machine learning algorithms to analyze data, build predictive models, and evaluate their performance using appropriate metrics.
- Teach students advanced dimensionality reduction methods to handle high-dimensional data, enhancing their ability to visualize, interpret, and preprocess data for machine learning tasks.
- Guide students in understanding and implementing artificial neural networks and deep learning techniques to solve complex real-world problems, emphasizing hands-on experience with modern tools and libraries.

Course Content	TEACHING HOURS
UNIT 1: Clustering Techniques	**14 Hrs.**
Introduction to Clustering: Definition, types of clustering (hard vs. soft), applications, and importance. **K-Means and Variants**: K-means algorithm, choosing the number of clusters (elbow method), K-means++, and limitations. **Hierarchical Clustering**: Agglomerative and divisive methods, dendrograms, linkage methods (single, complete, average), and practical applications. **Density-Based Clustering**: DBSCAN, key parameters (epsilon, minPts), and comparison with K-means and hierarchical methods.	
UNIT 2: Classification Techniques	**14 Hrs.**
Introduction to Classification: Overview, types of classification problems, binary vs. multi-class classification. **Bayesian Classifiers**: Naive Bayes, assumptions, advantages, limitations, and Bayesian networks. **Decision Trees and Random Forests**: Concept of decision trees, information gain, Gini index, overfitting, pruning techniques, and introduction to Random Forests. **Support Vector Machines (SVM)**: SVM for linearly separable data, kernel methods for non-linearly separable data, hyperplane and margin concepts. **K-Nearest Neighbors (KNN)**: KNN algorithm, choice of K, distance metrics, and performance optimization.	
UNIT 3: Dimensionality Reduction Techniques	**14 Hrs.**

Introduction to Dimensionality Reduction: Importance, challenges of high-dimensional data, and the curse of dimensionality. **Principal Component Analysis (PCA):** Eigenvalues, eigenvectors, explained variance, and interpretation of PCA components. **Linear Discriminant Analysis (LDA):** Fisher's criterion, maximizing class separability, and LDA vs. PCA. **Feature Selection Methods:** Filter methods, wrapper methods, and embedded methods.	
UNIT 4: Ensemble Learning Methods	**14 Hrs.**
Ensemble Learning: Definition and motivation for ensemble methods, Types of ensemble methods, Advantages of ensemble learning over single models. **Bagging and Random Forests:** Bootstrap Aggregating (Bagging) concept, Random Forests: construction, feature selection, and out-of-bag error estimation, Comparison of Random Forests with Decision Trees. **Boosting Techniques:** Overview of boosting, AdaBoost: algorithm, weight updates, and practical considerations, Gradient Boosting Machines (GBM): concept, learning rate, and overfitting prevention.	
Textbooks 1. Introduction to Machine Learning by Ethem Alpaydin, MIT Press 4th Edition (2020) 2. Pattern Classification by Duda and Hart. John Wiley publication 2nd Edition. 3. Tom M. Mitchell, "Machine Learning", McGraw-Hill, 2010	
Reference Books 1. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer 1st Edition (2006) 2. Machine Learning: A probabilistic Perspective, by Kevin P. Murphy, MIT Press 1st Edition (2012) 3. Introduction to Machine Learning by Ethem Alpaydin, MIT Press, 4th Edition (2020) 4. Pattern Classification by Duda and Hart. John Wiley publication, 2nd Edition (2000) 5. The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2nd Edition (2009)	
COURSE OUTCOMES (CO): **CO1:** Students will demonstrate the ability to implement and evaluate various clustering and classification algorithms, including K-means, hierarchical clustering, decision trees, SVMs, and KNN, applying them effectively to real-world datasets. **CO2:** Students will acquire the skills to apply dimensionality reduction techniques like PCA, LDA, and t-SNE, optimizing models for better performance and interpretability in high-dimensional spaces. **CO3:** Students will be able to design, train, and optimize artificial neural networks, including deep learning architectures such as CNNs and RNNs, for applications in areas like image and speech recognition. **CO4:** Students will develop the expertise to analyze complex data-driven problems, design appropriate machine learning solutions, and critically evaluate their models using rigorous validation techniques and metrics.	

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	3	1	-	-	2	1	-	2
2	3	3	2	2	3	1	-	-	1	1	-	2
3	3	3	3	3	3	2	-	1	2	1	1	3
4	3	3	3	3	3	1	1	1	2	2	2	3

COURSE TITLE: Cloud Computing

Course Code: MCA24209DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Understand core cloud computing concepts and service models.
- Gain practical skills in deploying and managing cloud applications.
- Understand how to manage cloud service performance, reliability, and security.
- Analyze the cost and benefits of different cloud platforms.

Course Content	TEACHING HOURS
UNIT 1: CLOUD COMPUTING FUNDAMENTALS	**14 Hrs.**
Cloud Computing definition;, private, public and hybrid cloud. Cloud types; IaaS, PaaS, SaaS. Benefits and challerges of cloud computing, public vs private clouds, Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery; next generation Cloud Applications.	
UNIT 2: VIRTUALIZATION AND CLOUD APPLICATIONS	**14 Hrs.**
VIRTUALIZATION: Role of virtualization in enabling the cloud : Types of Virtual Machines, Advantages of Virtualization, Components of Virtualization, CLOUD APPLICATIONS : Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages	
UNIT 3: MANAGEMENT OF CLOUD SERVICES	**14 Hrs.**
Reliability, availability and security of services deployed from the cloud. Performance and scalability of services, tools and technologies used to manage cloud services deployment; Cloud Economics: Cloud Computing infrastructures available for implementing cloud based services. Economics of choosing a Cloud platform for an organization, based on application requirements, economic constraints and business needs (e.g Amazon, Microsoft and Google, Salesforce.com, Ubunta and Redhat)	
UNIT 4: APPLICATION DEVELOPMENT	**14 Hrs.**
Application Development: Design and implementation in cloud environments. Development Platforms: AWS, Azure, Google App Engine. Deployment and management strategies for cloud applications.	

Textbooks

- Gautam Shroff, "Enterprise Cloud Computing: Technology, Architecture, Applications", Cambridge University Press; 2nd Edition [ISBN: 9780521137355], 2023.
- Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach" McGraw-Hill Education; 2nd Edition [ISBN: 9780071826400], 2018.
- Dimitris N. Chorafas, "Cloud Computing Strategies" CRC Press; 2nd Edition [ISBN: 9780367338611], 2021.

Reference Books

- Thomas Erl, "Cloud Computing: Concepts, Technology & Architecture" Prentice Hall; 3rd Edition [ISBN: 9780133994164], 2024.
- Rajkumar Buyya, Christian Vecchiola, and Selvi, S. Thamarai, "Mastering Cloud Computing: Foundations and Applications Programming" Morgan Kaufmann; 3rd Edition [ISBN: 9780128180747], 2022.

COURSE OUTCOMES (CO):

CO1: Explain cloud computing principles and service models.
CO2: Successfully deploy and manage cloud-based applications.
CO3: Apply best practices for cloud service management.
CO4: Assess the economic aspects of cloud computing platforms

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Linux Programming

Course Code: MCA24210DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Describe the structure, features and utilities available in Linux
- Use Linux utilities for system administration
- Develop basic applications using Shell scripting
- Describe various methods of extending a Linux kernel
- Develop kernel modules for extending Linux kernel
- Develop GUI applications using Qt programming

Course Content	TEACHING HOURS
UNIT 1: Introduction to Linux	**14 Hrs.**
Introduction – History, acquisition and installation, Linux features and directory structure. Linux utilities – directory and file manipulation, text processing, process management, system information, creating and managing users, setting ownerships/permissions, managing services.	
UNIT 2: Shell scripting	**14 Hrs.**
Shell – definition & types. Variables – local, shell & environment. Operators – test, expr, bc, built-in. Floating-point arithmetic. Expressions – arithmetic, relational and logical. Looping & decision-making statements. Substitution – filename, variable and command. Functions and positional parameters. Writing shell scripts for developing basic applications.	
UNIT 3: Kernel development	**14 Hrs.**
Linux kernel architecture. Building the kernel. Extending the kernel – Syscalls and kernel modules. Compiling Modules. Loading/unloading modules. Module licensing. Exporting symbols. Writing kernel modules for extending Linux kernel.	
UNIT 4: GUI programming	**14 Hrs.**
X Window System - Introduction, history, features and working. X-Server, X-Protocol, X-Client, & X-lib. Qt toolkit – Introduction, cross-platform GUI development. Qt creator. Basic structure of a Qt program. Compilation. Signal-Slot mechanism. Qt widgets. Container widgets. Custom layouts and slots. Writing Qt programs for developing basic GUI applications.	

Textbooks:

1. R. Petersen, LINUX: The Complete Reference, 6th Edition, Tata McGraw Hill, 2008.

Reference Books:

1. S. Veeraraghavan. Shell Programming in 24 hours. SAMS/Techmedia, 2007.
2. R. Love. Linux Kernel Development. Addison-Wesley, 2010.
3. J. Blanchette, M. Summerfield. C++ GUI Programming with Qt3. Prentice Hall, 2004.

COURSE OUTCOMES (CO):

CO1: Students will be able to describe the structure, features and utilities available in Linux
CO2: Students will be able to use Linux utilities for system administration
CO3: Students will be able to develop basic applications using Shell scripting
CO4: Students will be able to describe various methods of extending a Linux kernel
CO5: Students will be able to develop kernel modules for extending Linux kernel
CO6: Students will be able to develop GUI applications using Qt programming

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Theory of Computation

Course Code: MCA24211DCE						**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	80
						Internal	20
Lecture (L):	4	**Practical (P):**	-	**Tutorial (T):**	-	**Total Credits**	4

Course Objectives:

- To Understand computational models and finite automata in formal language theory and computational complexity.
- To Design and analyze DFA and NFA, understand regular languages, and their equivalence with regular expressions.
- To Study context-free languages (CFLs), grammars (CFGs), parse trees, and pushdown automata (PDA).
- To Explore context-sensitive languages (CSL), linear bounded automata (LBA), recursive languages (REL), and Turing machines (TM).
- To Learn about decidability, undecidability, reduction techniques, and complexity theory foundations.

COURSE CONTENT	TEACHING HOURS
UNIT 1: Introduction to Computation	**14 Hrs.**
Introduction to computation, Regular Languages: Introduction to formal languages, regular operations, Closure property. Finite Automata, Deterministic Finite Automata, Kleene's theorem, Non-deterministic Finite Automata (NFA), ε-NFA, Conversion of ε-NFA to NFA, NFA to DFA, Minimization, Finite Automata with output: Mealy and Moore machines. Regular Expression; Equivalence of DFA, NFA, and RE. Non-Regular Languages and Pumping Lemma.	
UNIT 2: Context-Free Languages	**14 Hrs.**
Introduction to Context-Free Languages (CFL), Pushdown Automata (PDA), Grammars, Context Free Grammars, Parsing and Ambiguity, Parsing and Membership, Inherent Ambiguity of Context-Free Languages, Chomsky Normal Form, Membership Algorithm for CFG. Deterministic vs non-deterministic PDAs. Closure property and Pumping Lemma for CFLs.	
UNIT 3: Context-Sensitive Languages and Turing Machine	**14 Hrs.**
Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Languages (CSL), Context Sensitive Grammars, Linear Bounded Automata (LBA). Introduction to Turing Machines, Turing Machines as Language Acceptors and Transducers, Turing's Thesis, Equivalence of Deterministic, Non-deterministic, and multi-tape TMs. Universal TMs.	
UNIT 4: Undecidability and Computational Complexity	**14 Hrs.**
Decidability and Undecidability, Reductions and its applications, Rice's theorems for RE sets, Post Correspondence Problem. Halting Problem, Halting vs Looping. Hilbert's algorithm. Complexity Classes (P and NP), Satisfiability (SAT) Problem, Hamiltonian Path Problem, Clique Problem. Polynomial Time Reduction.	

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-determinaistic 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

	POs											
COs	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)

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

	Pos											
COs	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