Department of Computer Science

Department of Computer Science
Islamic University of Science & Technology

Course Curriculum
(Choice Based Credit System)

for

Master of Computer Applications (MCA)
Two Year Programme
2023 and Onwards

CBCS Course Outline MCA

List of Abbreviations

SEC	“Skill Enhancement Course”
FC	“Foundation Course”
VAC	“Value Added Course”
AEC	“Ability Enhancement Course”
MDC	“Multidisciplinary Course”
EC	“Employability Course”
EOC	“Entrepreneurship Oriented Course”

Course Outline MCA Semester I

Semester-I (24 Credit Semester with 31 hours per week)								
Course Code	**Course Name**	**Paper category**	**Total Credit**	**Credit Hours**			**Hours per week**	**Course Type**
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA501C	Programming Concepts in C/C++	Core	5	3	0	4	07	SEC
MCA502C	Database Management System	Core	5	3	0	4	07	FC
MCA503C	Discrete Mathematics	Core	4	3	1	0	04	FC
MCA504C	Computer Organization and Architecture	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MCA511E	Digital Electronics	DCE	3	2	1	0	03	SEC
MCA512E	Programming Languages & Paradigm	DCE	3	2	1	0	03	SEC
MCA513E	Management Information System	DCE	3	2	1	0	03	EC
MCA514E	Elements of Business Management	DCE	3	2	1	0	03	EOC
Pool B: 3 Discipline Centric Elective Credit Units								
MCA526E	Web Technologies	DCE	3	2	0	2	04	EOC
MCA527E	E-Commerce	DCE	3	2	0	2	04	EC
MCA528E	Multimedia & Image Authoring	DCE	3	2	0	2	04	EC
MCA529E	Programming in R	DCE	3	2	0	2	04	SEC
Audit Courses (Mandatory)								
MCA541A	Information and Communications Technology	Audit	0	2	0	0	02	AEC
Total			**24**				**31**	

Course Outline MCA Semester II

Semester-II (26 Credit Semester with 33 hours per week)								
Course Code	**Course Name**	**Paper category**	**Total Credit**	**Credit Hours**			**Hours per week**	**Course Type**
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA551C	Data Structures	Core	5	3	0	4	07	FC
MCA552C	Artificial Intelligence	Core	5	3	0	4	07	AEC
MCA553C	Software Engineering	Core	4	3	1	0	04	FC
MCA554C	Operating System	Core	4	3	1	0	04	FC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MCA561E	Advanced Computer Organization & Architecture	DCE	3	2	1	0	03	AEC
MCA562E	Operational Research	DCE	3	2	1	0	03	AEC
MCA563E	Microprocessor & Assembly Language Programming	DCE	3	2	1	0	03	AEC
MCA564E	Soft Computing	DCE	3	2	1	0	03	SEC
Pool B: 03 Discipline Centric Elective Credit Units								
MCA576E	Modeling & Simulation	DCE	3	2	0	2	04	SEC
MCA577E	Computer Graphics	DCE	3	2	0	2	04	SEC
MCA578E	Numerical and Statistical Computing	DCE	3	2	0	2	04	VAC
MCA579E	Programming in Python	DCE	3	2	0	2	04	SEC
MCA580E	Java Programming	DCE	3	2	0	2	04	EC
Audit Courses (Mandatory)								
MCA591A	Technical Communication	Audit	0	2	0	0	02	VAC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			**26**				**33**	

**** to be chosen from other offering Departments of IUST**

Course Outline MCA Semester III

Semester-III (26 Credit Semester with 31 hours per week)								
Course Code	**Course Name**	**Paper category**	**Total Credit**	**Credit Hours**			**Hours per week**	**Course Type**
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA601C	Full-Stack Development	Core	5	3	0	4	07	EC
MCA602C	Machine Learning	Core	5	3	0	4	07	AEC
MCA603C	Data Communication & Computer Networks	Core	4	3	1	0	04	FC
MCA604C	Analysis and Design of Algorithms	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MCA611E	Advanced Operating Systems	DCE	3	2	1	0	03	AEC
MCA612E	Distributed Databases	DCE	3	2	1	0	03	SEC
MCA613E	Bioinformatics	DCE	3	2	1	0	03	AEC
MCA614E	System Software Design	DCE	3	2	1	0	03	AEC
MCA615E	Natural Language Processing	DCE	3	2	1	0	03	AEC
Pool B: 3 Discipline Centric Elective Credit Units								
MCA626E	Digital Image Processing	DCE	3	2	0	2	04	AEC
MCA627E	Linux System Programming	DCE	3	2	0	2	04	SEC
MCA628E	Data Warehousing	DCE	3	2	0	2	04	EC
MCA629E	Open Source Technologies	DCE	3	2	0	2	04	EOC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			**26**				**31**	

**** to be chosen from other offering Departments of IUST**

Course Outline MCA Semester IV

Semester-IV (24 Credit Semester and 32 hours per week)								
Course Code	**Course Name**	**Paper category**	**Total Credit**	**Credit Hours**			**Hours per week**	**Course Type**
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA651C	Theory of Formal Languages	Core	4	3	1	0	04	FC
MCA652C	Big Data Analytics	Core	4	3	1	0	04	AEC
MCA653C	Project work	Core	10	1	2	7	17	EOC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MCA661E	Wireless Communications	DCE	3	2	1	0	03	SEC
MCA662E	Cloud and Grid Computing	DCE	3	2	1	0	03	SEC
MCA663E	Information Security and Networks	DCE	3	2	1	0	03	SEC
MCA664E	Compiler Design	DCE	3	2	1	0	03	AEC
Pool B: 03 Discipline Centric Elective Credit Units								
MCA676E	Block Chain Technologies	DCE	3	2	0	2	04	EC
MCA677E	Quantum Computing	DCE	3	2	0	2	04	AEC
MCA678E	Advanced Java	DCE	3	2	0	2	04	EC
MCA679E	Deep Learning	DCE	3	2	0	2	04	SEC
Total			**24**				**32**	

Semester - I

Semester-I (24 Credit Semester with 31 hours per week)

Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA501C	Programming Concepts in C/C++	Core	5	3	0	4	07	SEC
MCA502C	Database Management System	Core	5	3	0	4	07	FC
MCA503C	Discrete Mathematics	Core	4	3	1	0	04	FC
MCA504C	Computer Organization and Architecture	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MCA511E	Digital Electronics	DCE	3	2	1	0	03	SEC
MCA512E	Programming Languages & Paradigm	DCE	3	2	1	0	03	SEC
MCA513E	Management Information System	DCE	3	2	1	0	03	EC
MCA514E	Elements of Business Management	DCE	3	2	1	0	03	EOC
Pool B: 3 Discipline Centric Elective Credit Units								
MCA526E	Web Technologies	DCE	3	2	0	2	04	EOC
MCA527E	E-Commerce	DCE	3	2	0	2	04	EC
MCA528E	Multimedia & Image Authoring	DCE	3	2	0	2	04	EC
MCA528E	Programming in R	DCE	3	2	0	2	04	SEC
Audit Courses (Mandatory)								
MCA541A	Information and Communication Technology	Audit	0	2	0	0	02	AEC
Total			**24**				**31**	

Course Title:	Programming Concepts in C/C++	**Course Code:**	MCA501C
Semester:	1st	**Paper Type:**	Core
Credits:	05	**Max Marks:**	125
Pre Requisite:	--------	**Co-Requisite:**	MCA503C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. Also by learning the basic programming constructs they can easily switch over to any other language in future.
- Develop an in-depth understanding of functional, logic, and object-oriented programming paradigms.
- To practice the fundamental programming methodologies in the C/C++ programming language via laboratory experiences.
- To code, document, test, and implement a well-structured, robust computer program using the C/C++ programming language.
- Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O and other standard language constructs.
- Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.

COURSE CONTENT:

UNIT I

C programming language: Evolution, Features & Importance. Basic Structure of C programs, Character Set, Identifiers, Reserved Words, Data Types, Constants, Variables, Symbolic Constants, Casting and Standard Libraries. Logical and Control Structures: Assignment, Arithmetic, Relational, Logical, Compound, Increment, Decrement, Bitwise Operators & Special Operators. IF, IF – ELSE, Nested IF – ELSE, ?: , SWITCH CASE. Looping Constructs: FOR, WHILE, DO-WHILE, EXIT, BREAK, CONTINUE

UNIT II

Arrays: Types of arrays, Initialization, dynamic arrays. Character Arrays & Strings. String-handling functions. Functions: Concepts, Elements, Prototypes & Types. Passing Arrays to Functions. Storage classes, Recursion. Command-line arguments. Multifile programming. Preprocessing. Pointers: Concepts, Variables, swapping data, swapping address v/s data, pointers & arrays.

UNIT III

Pointers to pointers, pointer to strings, pointer arithmetic, additional operators, pointers to functions, void pointers. Structures and Unions: Syntax & use, members, structures & pointers, array of structures, structures & functions, structure within structures. OOPS: Evolution and need of C++, Advantages over Procedural programming. Introduction to classes and objects, Basic OOPS programming.

UNIT IV

C++ Functions: passing arguments to functions, returning values from functions, reference arguments, inline functions, default arguments, object as function argument, returning objects from functions. Constructors and

Destructors, Copy Constructors. Inheritance and Polymorphism: Inheritance and types, Polymorphism (static and dynamic), function overloading, function overriding, virtual functions & operator overloading.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Formulate simple algorithms for arithmetic and logical problems, translate the algorithms to programs (in C language),
- Test and execute the programs and correct syntax and logical errors,
- Develop efficient algorithms for solving a problem and implementation.
- Use the various constructs of a programming language viz. conditional, iteration and Recursion.

Text Books:

1. Programming in ANSI C 6th Edition "E. Balaguruswamy"
2. Robert Lafore, "Object Orientation with C++ Programming", Waite Grou

References:

1. Object Oriented Programming with C++ " E. Balagurusamy"
2. Herbert Schildt, "C++ The Complete Reference", Tata McGraw Hill
3. Dennis Richie & Kernighan, "C Programming Language", Prentice Hall
4. Dietel & Dietel, "How to program", Pearson Education

List of Practicals:

LAB JOURNAL_MCA501C	
S. No.	**Program Title**
1.	WAP to perform various Arithmetic Operations like: i. Addition of two numbers. ii. Subtraction of two numbers. iii. Multiplication of two numbers. iv. Simple division of two numbers. v. Modular division of two numbers.
2.	WAP to check whether a number is even or odd.
3.	WAP to show the use of typedef.
4.	WAP to show the use of enum.
5.	WAP to show the use of Relational Operators (< , >, <=, >=, ==, !=)
6.	WAP to show the use of logical Operators like: i. Logical AND (&&) ii. Logical OR (||). iii. Logical NOT (!).
7.	WAP to show the use of Pre / Post Increment.
8.	WAP to show the use of Pre / Post Decrement.
9.	WAP to show the use of Ternary /Conditional operator.
10.	WAP to show the use of sizeof operator.

11.	WAP to show the use of Bitwise operators as: i. Bitwise OR. ii. Bitwise NOT. iii. Bitwise <<. iv. Bitwise >>.
12.	WAP to show the user of Implicit type conversion.
13.	WAP to show the use of Explicit type conversion.
14.	WAP to show the use of simple if (Greater among two numbers).
15.	WAP to show the use of simple if (Greater among any numbers).
16.	WAP to show the use of if-else (Salary increase if Salary>10000).
17.	WAP to show the use of else-if (Grading of Students).
18.	WAP to show the use of switch (Months of Year).
19.	WAP to show the use of switch (Days of Week).
20.	WAP to show the use of while loop (Sum of digits of a number).
21.	WAP to show the use of while loop (Reverse of a number).
22.	WAP to show the use of while loop (X raised power n).
23.	WAP to print Fibonacci series using while loop.
24.	WAP to print whether number is Prime or not using while loop.
25.	WAP to print table using while loop.
26.	WAP to print sum of odd series using while loop.
27.	WAP to print sum of even series using while loop.
28.	WAP to find factorial of any number using while loop.
29.	WAP to show the use of for loop (Sum of digits of a number).
30.	WAP to show the use of for loop (Reverse of a number).
31.	WAP to show the use of for loop (X raised to Power n).
32.	WAP to print Fibonacci series using for loop.
33.	WAP to print whether number is Prime or not using for loop.
34.	WAP to print table using for loop.
35.	WAP to print sum of odd series using for loop.
36.	WAP to print sum of even series using for loop.
37.	WAP to find factorial of any number using for loop
38.	WAP for calculating the sum of even /odd series upto ‘n’.
39.	WAP for converting a given number of days into months and remaining days.
40.	WAP to draw different structures like Triangle, Pyramid etc.
41.	WAP for calculating the sum of elements of array.
42.	WAP for obtaining the greatest element of array.
43.	WAP to insert an element into an array at different location.
44.	WAP for displaying an entered string.
45.	WAP to find frequency of elements in an array of ‘n’ elements.
46.	WAP for calculating string length. • Including spaces. • Excluding spaces.
47.	WAP to reverse the elements of an array of ‘n’ elements.
48.	WAP to delete an element in an array at a given location. • At start. • At some specific given position. • At the end of array.
49.	WAP for Addition of two Matrices.
50.	WAP for Subtraction of two Matrices.
51.	WAP for Multiplication of two Matrices.

52.	WAP for obtaining the transpose of a Matrix.
53.	WAP to implement a stack structure with its operations using arrays.
54.	WAP to implement a Queue structure with its consequent operations using arrays.
55.	WAP to find length of a string without using any string handling function.
56.	WAP to perform string concatenation, string comparison without using any string handling function.
57.	WAP to perform string concatenation, string comparison while using string handling functions.
58.	WAP to find length of a string using string handling function.
59.	WAP to search a particular word from a given string.
60.	WAP to show the use of two dimensional arrays.
61.	WAP to show the use of Multidimensional arrays.
62.	WAP to show the use of functions including function declaration, function call and function definition.
63.	WAP to show the use of various categories of functions.
64.	WAP to show pass by value and pass by reference.
65.	WAP to show how to pass an array as an argument to a function.
66.	Write down all programs ranging from program 41 to program 61 using functions.
67.	WAP to show recursion (Factorial of a number).
68.	WAP to show the use of Storage classes.
69.	WAP to show Multifile programming.
70.	WAP to show the use of Pointers.
71.	WAP to show usage of Structures.
72.	WAP to show Copying and Comparing Structures.
73.	WAP to show the use of Array of Structure variables.
74.	WAP to show the use of Arrays within Structures.
75.	WAP to show the use of Structures within structures.
76.	WAP to show Structures and Functions.
77.	WAP to show the use of Unions.
78.	WAP to show the use of Bit fields.
79.	WAP to show the use and understanding of Pointers (Declaring, Initializing and Accessing pointers).
80.	WAP to show the use and understanding of chain of Pointers.
81.	WAP to show the use of Pointer Expressions.
82.	WAP to show the use of Pointer Increments / Decrements and Scale Factor.
83.	WAP to show the use of Pointers and Arrays.
84.	WAP to show the use of Pointers and Character arrays.
85.	WAP to show the use of Array of Pointers.
86.	WAP to show the use of Pointers as Function arguments.
87.	WAP to show the use of Functions returning Pointers.
88.	WAP to show the use of Pointers and Structures.
89.	WAP to show the use of Classes.
90.	WAP to show the use of Functions in C++ (Function prototype)
91.	WAP on Inline Functions.
92.	WAP to use call by reference and return by reference.
93.	WAP on default arguments.
94.	WAP to show Function Overloading.
95.	WAP in C++ to show nesting of functions.
96.	WAP to show static data members and static member functions.
97.	WAP to show objects as function arguments and returning objects.
98.	WAP on default and reference arguments
99.	WAP to show array of objects, their declaration and their passing to a function.
100.	WAP on constructors.

101.	WAP on parameterized constructors.
102.	WAP on multiple constructors in a class.
103.	WAP on copy constructor and its use.
104.	WAP on destructors.
105.	WAP to show Inheritance in general.
106.	WAP to show single Inheritance.
107.	WAP to show Multiple Inheritance.
108.	WAP to show Multilevel Inheritance.
109.	WAP to show Hierarchical Inheritance.
110.	WAP to show Hybrid Inheritance.
111.	WAP on Operator Overloading.
112.	WAP to show this pointer.
113.	WAP to show virtual base class.
114.	WAP to show virtual functions and pure virtual functions
115.	WAP to show function overriding.
116.	WAP to show command line arguments.
117.	WAP to show the use of Multidimensional arrays.
118.	WAP to show the use of functions including function declaration, function call and function definition.
119.	WAP to show the use of various categories of functions.
120.	WAP to show pass by value and pass by reference.
121.	WAP to show how to pass an array as an argument to a function.
122.	Write down all programs ranging from program 41 to program 61 using functions.

Course Title:	Database Management System	**Course Code:**	MCA502C
Semester:	1st	**Paper Type:**	Core
Credits:	05	**Max Marks:**	125
Pre Requisite:	--------	**Co-Requisite:**	-------------

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand the role of a database management system in an Organization.
- To understand basic database concepts including the structure and Operation of the relational data model.
- To construct simple and moderately advanced database queries using Structured Query Language (SQL).
- To understand and successfully apply logical database design principles, including E-R diagrams and database normalization.

COURSE CONTENT:

UNIT I

Basic Concepts and Conceptual Database Design: Database Users, Characteristics of the Database, Advantage of using Database Systems, Data Models, schemas and instances, Three Tier Architecture & Data Independence, Database Languages & Interfaces. Overview of Legacy Data Base Management Systems. Data Modeling Using The Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak Entity Sets, Translating your ER Model into Relational Model.

UNIT II

Relational Model, Languages & Systems: Relational Data Model, Relational Model Concepts, Relational Model Constraints, Relational Algebra, SQL – A Relational Database Language, Data Definition & Manipulation in SQL, Queries in SQL, Specifying Constraints in SQL, Practicing SQL commands using ORACLE, "PL/SQL, Stored Procedures".

UNIT III

Functional Dependencies & Normalization for Relational Databases: Functional Dependencies, Normal Forms based on primary keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependencies. "Mini Project: Data Analysis and Data Modelling".

UNIT IV

Transaction Management: Transaction Concept and State, Desirable Properties of a Transaction, Characterizing Schedules based on Serializability and Recoverability, Concurrency Control Techniques: Lock-Based Protocols, Timestamp-based Protocols, Validation–based Protocols. Database Recovery Techniques: Recovery Concepts, Recovery based on Deferred Update and Immediate Update. Shadow Paging. Overview of Object Oriented Database Management Systems, Distributed Data Base Management Systems.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Introduces the role of a database management system, basic database concepts, including the structure and operation of the relational data model.
- Familiarize themselves with the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).

- Have knowledge of database transaction including concurrency control, backup and recovery, and data object locking.
- Design and implementation of a small database project using Oracle.

Text Books:

1. Korth, Silberschatz, "Database System Concepts", TMH
2. Elmsari & Navathe, "Fundamentals of Database Systmes", A. Wesley
3. Ullman J. D., "Principles of Database Systems", Galgotia Publications

References:

4. Steve Bobrowski, "Oracle 8 Architecture", TMH
5. Date C. J., "An Introduction to Database Systems",Narosa Publishing
6. William Page, "Using Oracle 8i – Special Edition", Que/PHI
7. Ivan Bayross, "SQL & PL/SQL Using Oracle 8i & 9i with SQLJ", BPB
8. Desai.B, "An introduction to Database Concepts", Galgotia Publications

List of Practicals:

LAB JOURNAL MCA_502C	
S. No.	**Program Title**
1.	Study of DBMS, RDBMS.
2.	To study Data Definition language Statements.
3.	To study Data Manipulation Statements.
4.	Study of SELECT command with different clauses.
5.	Study of SINGLE ROW functions (character, numeric, Data functions).
6.	Study of GROUP functions (avg, count, max, min, Sum).
7.	Study of various type of SET OPERATORS (Union, Intersect, Minus).
8.	Study of various type of Integrity Constraints.
9.	Study of Various type of JOINS. To create a DDL to perform creation of table, alter, modify and drop column.
10.	Define various Database Languages.
11.	Write a program to create a database.
12.	Write a program to create table with constraints such as 'NOT NULL', 'UNIQUE', 'DEFAULT', 'PRIMARY KEY', 'CHECK' etc.
13.	Write a program to add a few record to the database.
14.	Write a program to insert values into a specific column.
15.	Write a program to study the viewing commands (select, update) and executes the following queries: • Find the names of all students who study in MCA 3rd year. • Find the student names whose age is greater than 20 years.
16.	Write a program for following query to modify the structure of table using Alter or Delete command: • Add an attribute named 'Phone' to the table 'Student'. • Drop the attribute 'Gender' from the table 'Student'. • Delete the entries from the table 'Student' who left the course in between.
17.	WAP to Update the record using 'WHERE' clause and without using 'WHERE' clause.
18.	Write a program to update the record using Sub Query.
19.	Write a program to add and remove a 'FOREIGN KEY'.
20.	Write a program to order the records in ascending and descending order.

21.	Write a program to Group the records using ‘HAVING’ clause.
22.	Write programs for Aggregate functions such as ‘SUM’, ‘COUNT’, ‘AVG’, ‘MAX’, ‘MIN’.
23.	Write programs for various relational algebraic operations such as ‘UNION’, INTERSECT’, ‘EXCEPT’.
24.	Write a program for Join operation as ‘INNER JOIN’, ‘OUTER JOIN’, and ‘CROSS JOIN’.
25.	Consider the following relations for an order processing database application in a company. CUSTOMER (Cust #: int, Cname: string, City: string) ORDER (Order #: int, Odate: date, Cust #: int, Ord-Amt: int) ORDER-ITEM (Order #: int, Item #: int, qty: int) ITEM (Item #: int, Unit Price: int) SHIPMENT (Order #: int, Warehouse #: int, Ship-Date: date) WAREHOUSE (Warehouse #: int, City: string) i) Create the above tables by properly specifying the primary keys and the foreign keys. ii) Enter at least five tuples for each relation. iii) Produce a listing: CUSTNAME, NO_OF_ORDERS, and AVG_ORDER_AMT, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer. iv) List the Order# for the orders that were shipped from all the warehouses that the company has in a specific city. v) Demonstrate how you delete Item# 10 from the ITEM table and make that field null in the ORDER-ITEM table. vi) Generation of suitable reports & Create a suitable front end for querying and displaying the results.
26.	Consider the following database of student enrollement in courses and books adopted for each course. STUDENT (regno: string, name: string, major: string, bdate: int) COURSE (course#: int, cname: string, dept: string) ENROLL (regno : string , course#: int , sem : int , marks : int) BOOK_ADOPTION (course#: int, sem: int, book_isbn: int) TEXT (book_isbn: int, book-title: string, publisher: string, author: string). i) Create the above tables by properly specifying the primary keys and the foreign key. ii) Enter at least five tuples for each relation. iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department. iv) Produce a list of text books (include course #, book_isbn,book-title) in the alphabetical order for courses offered by the cs department that use more than 2 books. v) List any department that has all its adopted books published by specific publisher. vi) Generation of suitable reports.
27.	Consider the following relations for the details maintained by a book dealer. AUTHOR (Author-id: int, Name: string, City: string, Country: string) PUBLISHER (Publisher-id: int, Name: string, City: string, Country: string) CATALOG (Book-id: int, title: string, author-id: int, Publisher-id: int, Category-id: int, Year: int, Price: int) CATEGORY (Category-id: int, Description: string) ORDER-DETAILS (Order-no:int, Book-id: int, Quantity: int) i. Create the above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation. iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000. iv. Find the author of the book which has maximum sales. v. Demonstrate how you increase the price of books published by a specific publisher by 10%. vi. Generation of suitable reports. vii. Create a suitable front end for querying and displaying the results.

Course Title:	Discrete Mathematics	**Course Code:**	MCA503C
Semest3er:	1st	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	--------	**Co-Requisite:**	MCA501C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To familiarize students about set theory concepts, operations and uses in computer sciences.
- To explain concept of Discrete Structures which include Semigroups, Monoids, and Groups together with their uses in computer science and their corresponding structures in computer science
- To explain students concepts like Partially Ordered Sets, Lattices and Finite Boolean Algebra and use in Computer Science.
- Give student a knowledge about finite state Machines and possible modeling of process using finite state Machine concepts.

COURSE CONTENT:

UNIT I

Proposition, Logic, Truth tables, Propositional Equivalence, Logical Equivalence, Predicates & Quantifiers, Sets: operations on sets, Computer representation of sets, Functions: Domain, Range, One-to-One, Onto, Inverses & Composition, Cardinality of a Set, sequences & summations, The growth of functions . Methods of Proof: Different methods of proof, Direct Proof, Indirect Proof, Mathematical Induction for proving algorithms.

UNIT II

Discrete probability, Advanced Counting Techniques: Inclusion-Exclusion, Applications of inclusion-exclusion principle, recurrence relations, solving recurrence relation. Relations: Relations & their properties, Binary Relations, Equivalence relations, Diagraphs, Matrix representation of relations & digraphs, Computer representation of relations & digraphs, Transitive Closures, Warshall's Algorithm.

UNIT III

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 & Circuits, Eulerian & Hamiltonian Paths, Matrix representation of graphs. Graph Coloring. Cyclometric Complexity

UNIT IV

Trees: Rooted trees, Application of trees: Binary Search Trees, Decision Trees, Prefix Codes, Tree traversal, trees & sorting, spanning trees, minimal spanning trees. "Catalans Series (Number), B+ Trees, Red-Black Trees Finite Boolean algebra, Functions on Boolean algebra, Boolean functions as Boolean polynomials. Groups & applications: Subgroups, Semigroups, Monoids, Product & quotients of algebraic structures, Isomorphism, Homomorphism

COURSE OUTCOMES:

At the end of the course student will be able to:

- Develop mathematical and logical thinking
- Express real life problems it in terms of predicates, quantifiers, and logical connectives and obtained its solution
- Utilize the concepts of relations and functions to solve simple real life problems.

Text Books:

1. KENNETH H. ROSEN “Discrete Mathematics & Its Applications, Tata McGraw – Hill
2. Ralph P. Grimaldi, Discrete And Combinatorial Mathematics An Applied Introduction

References:

1. LIU “Elements of Discrete Mathematics “ Tata McGraw Hill
2. SCHAUMS “Discrete Mathematics“ Tata McGraw Hill
3. KOLMAN/REHMAN “Discrete Mathematical Structures“ Pearson Education
4. NICODEMI “Discrete Mathematics“ CBS

Course Title:	Computer Organization and Architecture	**Course Code:**	MCA504C
Semester:	1st	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	--------	**Co-Requisite:**	MCA501C,MCA511E

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To know and understand the main components of a computer system and the considerations in their design.
- To acquire tools for comparison among alternatives.
- To know and understand performance measures, as well as their impact on system architecture.
- To understand the interplay among system components, design trade-offs, etc.

COURSE CONTENT:

UNIT I

Fundamental concepts of computer architecture & organization, Register Organization. Interconnection Structures, Bus Interconnections, Integer/Floating Point Arithmetic & Representation, Instruction Cycle & Interrupts. Instruction Set Characteristics & Functions. Addressing Modes & Formats.

UNIT II

Processor Organization: ALU, Design of Arithmetic Circuit, Design of Logic Circuit & Design of ALU. Processor Unit: Design of Accumulator. Control Organization: Hardwired / Micro-Programmed Control, Control Memory, Address Sequencing, Design of Control Unit & Micro-Program Examples.

UNIT III

Memory Hierarchy, Main Memory: RAM/ROM Chips. Memory Address Map, Memory Connection to CPU, Associative Memory, Hardware Organization – Match Logic, Read/Write Operation, Cache Memory, Virtual Memory, Memory Management – Associated Hardware, I/O Organization. Peripheral Devices: I/O Interfaces, Asynchronous Data Transfer, Modes of Transfer, Direct Memory Access & I/O Processor.

UNIT IV

RISC Architectures – their characteristics & comparison to CISC, Introduction to Parallel Processing, Basic Parallelization Techniques. Pipelining – Arithmetic & Instruction Pipelining, RISC Pipeline, Vector & Array Processors. Multiprocessor System: Architectures & their Characteristics, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication & Synchronization, Cache Coherence Issues. Introduction to Super Scalar Processors

COURSE OUTCOMES:

At the end of the course student will be able to:

- Get familiarized with basics of computer hardware and how software interacts with computer hardware.
- Understand how computers represent and manipulate data, computer arithmetic and conversion between different number systems.
- Understand how Boolean algebra is related to designing computer logic, through simple combinational and sequential logic circuits.
- Understand basics of Instruction Set Architecture (ISA).

- Understand with a simple computer with hardware design including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow.
- Design combinational and sequential logic circuits, flip-flops, counters, shift registers, adders, subtractor, multiplexer, demultiplexer, Arithmetic/Logic unit.
- Understand concept of memory unit and input/output architecture.

Text Books:

1. W. Stallings, “computer organization & architecture”.

References:

1. V.C. Hamacher, A.G. Vranesicand S.G. Zaky, “Computer Organization”, Tata McGraw Hill
2. J.P Hayes, “Computer Architecture & Organization”, Tata Mcgraw Hill
3. Morris Mano, “Computer System Architecture”, PHI.
4. M. J. Flynn, “Computer Architecture”, Narosa
5. David A. Patterson, John L. Hannessey, ” Computer Organization”
6. GovaindaRajalu, “Computer Architecture & Design ” TMH

Course Title:	Digital Electronics	**Course Code:**	MCA511E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA504C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVE:

- To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
- To prepare students to perform the analysis and design of various digital electronic circuits.
- The course also covers digital logic design intended to make students familiar with different types of designs as sequential logic circuits and combinational logic circuits.

COURSE CONTENT:

UNIT I

Introduction to Digital Systems, Number systems: Binary number system, Octal & Hexa-decimal number system, Conversion of Number System, Arithmetic operations on Binary numbers, Number representation: Signed Binary numbers, 1's and 2's Complement. Fixed- and Floating-Point Representation. Logic Gates: AND, OR, NOT NOR, NAND & XOR & their Truth tables Logic diagrams. Boolean Postulates and Laws- Principal of Duality.

UNIT II

Minimization of Boolean Expression: Introduction, Minterm, Maxterm, Sum of Products (SOP)- Product of Sums (POS), Karnaugh map Minimization upto 4 variables. Don't Care Conditions. Error Detection and Correction: single parity and block parity check, hamming code, Combinational Logic: Introduction to Adders, Half Adder, Full Adder, Parallel binary adders, Subtractors- Half Subtractor, Full Subtractor, Multiplexer, De-multiplexer, Decoders, Encoder.

UNIT III

Sequential Logic - Definition & Basic sequential circuits, Latches & Flip-flops: SR-Flip Flop, D-Flip Flop, JK flip-flop, T flip-flop, Race condition, JK Master Slave Flipflop, Timing Specifications, Counters: Basics of Asynchronous & Synchronous counters, Registers, Shift Registers, Types of Shift registers.

COURSE OUTCOMES:

At the end of the course, students will be able to:

- To understand and examine the structure of various number systems and its application in digital design.
- Understand, analyze and design various combinational and sequential circuits.
- Identify basic requirements for a design application and propose a cost effective solution.
- Identify and prevent various hazards and timing problems in a digital design.
- To develop skill to build, and troubleshoot digital circuits

Text Books:

1. Digital Logic and Computer Design by M Morris Mano

References:

1. Digital Fundamentals, Global Edition (Kindle Edition) by Thomas L Floyd.
2. Modern Digital Electronics, 3rd Edition by R.P Jain
3. Digital Electronics: principles, Devices & Applications, Wiley by Anil K Maini.

Course Title:	Programming Languages & Paradigm	**Course Code:**	MCA512E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA501C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
- To introduce notations to describe syntax and semantics of programming languages.
- To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
- To introduce the concepts of ADT and object oriented programming for large scale software development.
- To design and extend operational and denotational definitions for basic programming language constructs.
- To prove properties of programs by various formal means, including structural and fix point induction.

COURSE CONTENT:

UNIT I

The role of Programming Languages: Towards Higher Level Languages programming paradigms, Language implementation. Language Description: Syntactic Structures, Expression Notations, Abstract Syntax trees, Lexical Syntax. Data Representation: The role of types, basic types, arrays, unions & variant records, Sets, Pointers, Two String tables, types & error checking. Procedure Activations: Introduction to Procedures, parameter passing methods, nested scope in source text, activation records, lexical scope: procedures as in C.

UNIT II

Object oriented Programming: class declarations in C++, dynamic allocation in C++, Information hiding. Functional Programming : Language of expressions , types, values & operations , approaches to expression evaluation, lexical scope, type checking, Function declaration by cases , Functions as first-class values, Implicit types, data types exception handling. Introduction to Prolog, data structures in Prolog, Programming techniques, controls in Prolog, Cuts

UNIT III

An introduction to concurrent Programming: Parallelism in hardware, Streams: implicit synchronization, concurrency as interleaving, Liveliness properties, safe accesses to shared data concurrency in ADA. Language Description: Semantic Methods, Synthesized attributes, Attribute grammars, natural semantics, Denotational Semantics.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Develop a greater understanding of the issues involved in programming language design and implementation
- Develop an in-depth understanding of functional, logic, and object-oriented programming paradigms
- Implement several programs in languages other than the one emphasized in the core curriculum (Java/C++)

- Understand design/implementation issues involved with variable allocation and binding, control flow, types, subroutines, parameter passing
- Develop an understanding of the compilation process

Text Books:

1. Ravi Sethi,“ Programming Languages ,Concepts & Constructs”, Pearson Education
2. Freidman, Wand, Haynes, “Essentials of Programming Languages”, PHI.

References:

1. Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education, 2008.
2. Programming languages – Watt, Wiley Dreamtech, 2004.
3. Programming Languages – Louden, Second Edition, Cengage,2003.
4. Programming languages – Ghezzi, 3/e, John Wiley, 1998.
5. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education, 2001.

Course Title:	Management Information System	**Course Code:**	MCA513E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA514E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The objective of Course Management Information System is to bring a systematic Knowledge of Management Information technology so that it can be appreciated and understood for application in business and industry, thus more emphasis is laid to application of information to business management and thus provides communication with employees, maintain employee records and coordinate work activities.

COURSE CONTENT:

UNIT I

Overview of Management Information Systems: Introduction, Concepts & characteristics, Components of MIS, Role of MIS, Concepts of Data, Information, Knowledge & Intelligence; Framework for Understanding MIS: Robert Anthony's hierarchy of management, MIS: Support to management, Relatedness of MIS with management activities, Attributes of Information system, Relationship between Organization & Information systems.

UNIT II

Introduction to decision making: Structured versus unstructured Decisions, Managerial Decision Making Process, Types of Decisions, Simons Model of decision making, Decision Support Systems: Overview, Concepts, Characteristics & Components, Overview of MBMS

UNIT III

Expert system: Basic Concepts, Comparison of conventional & expert systems, Structure of expert system, "Comparison of ES and DSS." Executive support system: Needs & characteristics, Role of executive support system in organization,

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the scope & importance of Management Information Systems
- Understand the concept and applications of DBMS, Systems Engineering Analysis and Design
- Understand & Apply Decision Support Systems Models with Digital Firm

Text Books:

1. Laudon, "Management Information Systems", Pearson
2. Jawadekar, "Management Infromation Systems",McGraw Hill Education India

References:

1. Kroenke, Management Information Systems", Mc-Graw Hill.
2. Mudrick R.G., Ross, J.E. andGleggt, J.R."Information Systems for Modern Management", PHI.
3. Jayashankar: Decision Support Systems, Mc-Graw Hill.
4. Stephen Haag & Maeve Cummings, "Information Systems Essentials", McGraw-Hill
5. Ralph Stair, "Principles of Information Systems", ISBN: 0619064897

Course Title:	Elements of Business and Management	**Course Code:**	MCA514E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA513E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand and analyze the environment of the organization.
- To help the students to develop cognizance of the importance of management principles.
- To help the students gain understanding of the functions and responsibilities of managers.
- To provide tools and techniques to be used in the performance of the managerial job.
- To understand process of decision problem formulation.

COURSE CONTENT:

UNIT I

Evolution of Management thought- Classical, Behavioral & Management Science, Managerial Process, functions, skills & roles in an organization, Levels of management, Planning concepts, process & parameters. Types of planning. The control process: concepts & significance, Importance of organization, formal organization elements; organizational chart, recruitment & selection; Sources of recruitment, selection criteria.

UNIT II

Motivation: meanings & approaches: carrot & stick approach , content theories of work motivation – Maslow's Need Hierarchy theory; Herzberg's motivation theory; Management control & audits; accounting audit. The management audit; purpose & scope.

UNIT III

Leadership: Meaning of leadership, Theories of Leadership; Trait theory; Situation theory; Path- Goal leadership Model; Leadership skills, Individual decision making & problem solving, Understanding & Managing group processing, Interpersonal & group dynamics

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the concepts related to Business.
- Understand the complexities associated with management of human resources in the organizations and integrate the learning in handling these complexities.
- `Demonstrate the roles, skills and functions of management.
- To formulate decision problems in the form of quantitative models.

Text Books:

1. Prasad L M, "principles & practices of management", SULTAN CHAND & SONS-NEW DELHI

References:

1. George R. Terry & Stephan G. Franklin, "Principles of Management".
2. Knootz, Harold & C.O. Dinell, "Management a system contingency analysis of managerial functions".

Course Title:	Web Technologies	**Course Code:**	MCA526E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA501C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To provide knowledge on web architecture, web services, client side and server side scripting technologies
- To focus on the development of web-based information systems and web services.
- To design interactive web pages using HTML and Style sheets.
- To provide skills to design interactive and dynamic web sites.
- To provide knowledge for implementing web applications using JavaScript.

COURSE CONTENT:

UNIT I

Web Essentials: Clients, Servers, and Communication. The Internet-Basic Internet Protocols, Introduction, Concept of Internet- History of Internet, Protocols of Internet, World Wide Web,Web browsers, web servers, MIME, URL, HTTP , URL, Web Clients ,Web Server , Web sites, Web Applications, types of Web applications,HTTP request-response message.

UNIT II

HTML- Introduction, History of HTML, Versions-Basic XHTML Syntax and Semantics, Structure of HTML Document: HTML Tags, open and closed tags, HTML Editors, Elements, Attributes, Styles, Formatting, working with formatting, form, tables, fames, links, images, lists, multimedia: Audio, video, Graphics: Canvas and SVG, HTML APIs: Geolocation, Drag/Drop, web storage.

UNIT III

Cascading Style Sheets: Introduction, syntax, selectors, colors, backgrounds, borders, Margins, padding, outline, text, Fonts, Icons, Lists, Tables, Display, Max-width, position, CSS Responsive: Viewport, Grid-view, media Queries, Images, video, frameworks and templates. Introduction to Javascript, controls statements, Arrays and functions, pattern matching, Element Access, Event Handling

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand & Apply (X)HTML(5)+CSS programming
- Demonstrate dynamic webpage development using java script and DHTML
- Design a well formed / valid XML document
- Create a server side ASP application using database

Text Books:

1. HTML5 Black Book by Dreamtech
2. HTML & CSS: Design and Build Web Sites by Jon Duckett,Wiley
3. Web Programming By Chris Bates , Wiley Publications

References:

1. Mastering HTML, CSS & Javascript Web Publishing by Laura Lemay, BPB Publications
2. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, by Dreamtech Press
3. HTML & CSS: The Complete Reference, Fifth Edition by Thomas Powell, McGraw Hill Education

List of Practicals:

LAB JOURNAL_MCA526E	
S. No.	**TITLE**
1.	Create a simple webpage using HTML.
2.	Basic HTML Tags, Table Tags, List Tags, Image Tags, Forms.
3.	Use new HTML5 form elements Email, Password, Files, Date, Number, Data list
4.	Include Audio, Videos in HTML pages
5.	Experimenting with CANVAS
6.	Drag and Drop in Webpages
7.	Incorporating Messaging feature in Websites;
8.	Add a Cascading Style sheet for designing the web page.
9.	Adding inline, internal and external CSS
10.	CSS Transitions
11.	CSS Animation
12.	Margin, Padding and Borders in CSS
13.	Background images in CSS
14.	Design a dynamic web page with validation using JavaScript.
15.	Use user defined function to get array of values and sort them in ascending order
16.	Demonstrate String and Math Object's predefined methods
17.	Demonstrate Array Objects and Date Object's predefined methods
18.	Exception Handling
19.	Calendar Creation : Display all month
20.	Write an HTML page that contains a selection box with a list of 5 countries In the above page when the user selects a country, its capital should be printed next to the list, and add CSS to customize the properties of the font of the capital.
21.	A web application that takes name and age from an HTML page. If the age is less than 18, it should send a page with "hello <name>,and u are not authorized to visit this site" otherwise it should send "welcome <name> to this site" message where name should be replaced with the entered name otherwise it should send welcome<name> to the site message
22.	WAP in JS to accept data from the user and display the same
23.	WAP in JS to demonstrate various I/O functions.
24.	WAP in JS to demonstrate the type conversion
25.	WAP in JS to perform various string manipulations
26.	WAP in JS to demonstrate the use of Objects, Arrays, Maps, Sets
27.	WAP in JS to create an animation
28.	WAP in JS to test if the number is even or odd
29.	WAP in JS to demonstrate the usage of conditional constructs
30.	WAP in JS to demonstrate the implementation of looping constructs
31.	WAP in JS to demonstrate the implementation of bitwise operators
32.	WAP in JS to demonstrate the implementation of logical operators
33.	WAP in JS to find the factorial of a number
34.	WAP in JS to check if a number is prime or composite

Course Title:	E-Commerce	**Course Code:**	MCA527E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA526E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To provide basic concepts of e-business and e-commerce, including presentation and discussion of the strategies and technologies involved.
- To provide theoretical and practical issues of conducting business over the internet and the Web
- To present methods for evaluating user needs.
- To provide an understanding of E-business Infrastructure, Selling and Marketing on the Web, Web Server Hardware and Software, Business-to-Business strategies, Virtual Communities, Web Portals, E-commerce Software, Payment systems, Security and User Experience.

COURSE CONTENT:

UNIT I

E-Commerce (Introduction and Definition), Goals of E-Commerce, Technical Components
Functions, Advantages and Disadvantages, Applications.

The Internet and WWW - Evolution of Internet, Domain Names and Internet Organisation (.edu, .com, mil-gov,.net ete), Internet Service provider.

UNIT II

E commerce business models: key elements of business model, Business to Consumer (B2C) model, B2B model, consumer to consumer model(C2C).Building E commerce- system development life Cycle, choosing software and hardware e commerce, site tools. Benefits of website, Registering a Domain Name, Web promotion. Internet Security, Secure Transaction, Computer Crime (types of Crimes), Threats.

UNIT III

Electronic Data Interchange, Introduction, Concepts of EDI and Limitation, Application of EDl, Disadvantages of EDI, EDI model, Electronic payment System, Introduction, Types, Strategies for developing electronic commerce web sites, Net marketplaces-characteristics of net marketplaces, types of net marketplaces, E distributors, E procurement, Exchanges. Online content providers-digital copyrights and electronic publishing

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the basic business management concepts, technical concepts, legal issues, and privacy relating to E-commerce.
- Understand how E-commerce is affecting business enterprises, governments, consumers and people in general.
- Describe the infrastructure needed for E-commerce and various electronic payment systems.

Text Books:

1. E-Commerce Concepts, Models, Strategies by G.S.V Murthy
2. E commerce: by Laudon

References:

1. E-Commerce by Kamlesh K Bajaj and Debjani Nag
2. Electronic Commerce by Gary P. Schneider

List of Practicals:

LAB JOURNAL_MCA527E	
S. No.	**TITLE**
1.	Perform a case study to understand the basic framework of e-commerce working environment.
2.	Perform a case study to understand the working paradigm of Amazon.in
3.	Perform a case study to understand the working paradigm of Flipkart.in
4.	Perform a case study to understand the working paradigm of Myntra.in
5.	Perform a case study to understand the working paradigm of Firstcry.in
6.	Perform a case study to understand the working of payment gateway in e-commerce.
7.	Perform a case study to understand the working of e-marketplace.
8.	Design a draft document to propose the basic requirements both technical and non-technical for establishing on line Grocery Store

Course Title:	Multimedia & Image Authoring	**Course Code:**	MCA528E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA526E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The main objective of the course is to learn and master the necessary skills in order to
- apply the most advanced technologies in computer graphics and multimedia systems, so that the students, will be able to decide the best techniques to be used or implemented in the creation, design and implementation of a graphical or multimedia application.
- Explore the Photoshop interface and use several tools for selecting parts of images.
- Learn to use layers and to apply layer effects and filters to create special effects. Additionally, use painting tools and blending modes to create shading effects.

COURSE CONTENT:

UNIT I

Multimedia System Design: Concept of multimedia, Multimedia Applications, Elements of Multimedia (Graphics/Images, Text, Audio, Video), Hypermedia vs hypertext, Overview of Multimedia authoring , authoring tools: Adobe Photoshop, Macromedia Flash, Dreamweaver, Windows Movie Maker. Study of File formats: Sound (AIFF, MIDI, WAV, MP3, ASF), Video (MPEG/MPG, AVI, MP4, 3GP, FLV). Image File Formats: TIFF, BMP, JPG/JPEG, GIF, PSD.

UNIT II

Image Editing Software (Adobe Photoshop) Basic Concepts: An Introduction, Menus, Toolbox, Color control icons, Mode control icons, Window controls icons, canvas, layers.

Layers: Adding layers, dragging and pasting selections on to layers, dragging layers between

files, viewing and hiding layers, Editing layers, rotating selections, scaling an object, preserving layers transparency, moving and copying layers, duplicating layers, deleting layers, merging layers, using adjustment layers.

UNIT III

Channels and Masks: Channel palette, showing and hiding channels, splitting channels in to separate image, merging channels, creating a quick mask, editing masks using quick mask mode; Painting and Editing: Brushes palette, brush shape, creating and deleting brushes, creating custom brushes, setting brush options, saving, loading and appending brushes, Options palette; Opacity, pressure, or exposure, paint fade-out rate, Using Selection tool.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the basic of multimedia technologies.
- Understand the process of creation & Implementation of various multimedia applications.
- Ability to use photoshop for various image effects.

Text Books:

1. Fundamentals of multimedia By Ze-Nian li and Mark s.Drew,Simon
2. Inside Adobe Photoshop 6-Bouton,Tay Vaughan. Tata McGraw Hill

References:

1. Andleigh PK and Thakrar K, “Multimedia Systems”, Addison Wesley Longman, 1999.
2. Fred Halsall, “Multimedia Communications”, Addison Wesley, 2000.
3. Fundamentals of Multimedia By Ze-Nian Li and Mark S. Drew , Simon Fraser University,
4. Pearson Education International.
5. Mastering CorelDraw 9 – Altman (BPB)

List of Practicals:

LAB JOURNAL_MCA528E	
S. No.	**TITLE**
9.	Perform a case study to understand the basic of images.
10.	Create images in various formats
11.	Processing images to convert between different formats
12.	Perform cropping operation on an image to remove unwanted elements or to improve composition.
13.	Increase/Decrease brightness and contrast to enhance the overall look of an image.
14.	Combine multiple images to create a composite image
15.	Editing image with quick Mask.
16.	Add special effects like lens flares, smoke, or rain to enhance the mood of an image.

Course Title:	Programming in R	**Course Code:**	MCA529E
Semester:	1st	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	--------	**Co-Requisite:**	MCA501C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To introduce programming with R.
- To introduce how to read data from files.
- To introduce how to perform excel-like operations with R.

COURSE CONTENT:

UNIT I

Introduction to R- Reserved Words, Variables & Constants, R Operators and Operator Precedence. Decision and Loop: if…else, for loop, while loop, break & next Functions: What are they? How to write your own Function, Function Return Value, Environment & Scope.

UNIT II

Vectors, Matrices and Lists. Data Frames and Factors. Slicing, Selection and Filtering. Arrays Basic Graphs & Charts: Bar Plot, Histogram, Pie Chart, Box Plot.

UNIT III

File Reading and Writing: How to read from and save to text files, csv files and excel sheets. Data Manipulation -Dealing with Missing and Duplicate Values, Sorting and Data Type Conversion, Merging and Joining Data Frames.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand R syntax, basic control flow and program design;
- Understand R data structures and how/when to use them;
- Understand how to perform Excel-like operations with R.

Text Books:

1. Cotton, R., Learning R: a step by step function guide to data analysis. 1st edition. O'reilly Media Inc
2. R PROGRAMMING FOR BEGINNERS by Sandhya Arora and Latesh Malik

References:;

1. The Art of R Programming: A Tour of Statistical Software Design 1st Edition, Kindle Edition by Norman Matloff (Author)
2. Statistical Programming in R by K.G. Srinivasa, G.M. Siddesh, et al.

List of Practicals:

LAB JOURNAL_MCA529E	
S. No.	**TITLE**
1.	Use of various data types in R
2.	Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric vectors.
3.	Write a program to show use of various control structures.
4.	Write a program to show use various types of function.
5.	Write a program to manipulate vectors, lists & Matricies.
6.	Extract substrings or manipulate character strings using functions like substr(), toupper(), or tolower().
7.	Create logical vectors using the c() function or by directly assigning TRUE or FALSE.
8.	Perform logical operations (AND, OR, NOT) using operators like &, |, and !.
9.	Use logical operators in control structures like if statements and loops.
10.	Create factors using the factor() function.
11.	Create lists using the list() function, containing elements of different data types.
12.	Access list elements using indexing ([[ ]]) or by name.
13.	Write a program to show use various operation on dataframe.
14.	Create data frames using the data.frame() function or by importing data from external sources.
15.	Explore the structure of data frames using functions like str(), head(), and summary().
16.	Handling Various array operations
17.	Creation of various charts & Graphs.
18.	Reading data from & writing data to files.

Course Title:	Information and Communications Technology	**Course Code:**	MCA541A
Semester:	1st	**Paper Type:**	Audit
Credits:	00	**Max Marks::**	50
Pre Requisite:	--------	**Co-Requisite:**	MCA526E, MCA512E

Marks Distribution: (Mid Term:15, End Term:25, Viva:05, Assignment / Presentations:05)

COURSE OBJECTIVE:

- The course introduces the students to basic computer concepts. Emphasis of the course is on providing the students with an introduction to programming, programming paradigms, database, basic networking and security.
- This course also covers digital logic design intended to make students familiar with different types of designs as sequential logic circuits and combinational logic circuits. Widely used software's including word processing, spreadsheets and presentations are studied.
- To build an appreciation for the fundamental concepts of computers and to become familiar with popular PC productivity software.

COURSE CONTENT:

UNIT I

Introduction to Computer: Generations and Characteristics.Components: Hardware – Input devices (OMR, OCR, MICR, etc.). Output devices, Central Processing Unit, Memory- Primary and Secondary. Software - Introduction, Types – System and Application. Operating system: Definition, Functions, Types, Classification, Elements of command based and GUI based operating system.Computer Network: Definition, criteria, process of communication, Types (LAN, WAN and MAN), Network topologies, devices. Transmission media: guided and unguided media.

UNIT II

Problem solving concept: Algorithms – Definition, Characteristics, Limitations. Computer Arithmetic & Number System: Positional number system – Binary, Octal and hexadecimal number systems; Methods of base conversions; Computer Arithmetic. Database: Overview, features, design, Database models –RDBMS.Internet: erview, Architecture, Functioning, Basic services like WWW, E-mail, Web Browsers.Overview of Emerging Technologies: Artificial intelligence, Big-data analytics, Cloud computing, Internet of Things (IoT), Block chain technology. Role of IT in different areas – Education, industry, banking, marketing, healthcare.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the basic concepts of computer and its component
- Relate theory and practice of computer architecture
- Illustrate the basic data representation in the computer
- Comprehend the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit

Text Books:

1. Rajaraman V., "Fundamentals of Computers", Prentice-Hall of India.
2. Norton P., "Introduction to Computers", McGraw Hill Education.
3. Goel A., "Computer Fundamentals", Pearson.
4. Balagurusamy E., " Fundamentals of Computers", McGraw Hill.
5. Thareja R., "Fundamentals of Computers", Oxford University Press.

6. Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it”, Penguin.

References:

1. V. Rajaraman, “Introduction to Computers”, TMH
2. Peter Norton, “Introduction to Computers”,TMH
3. Suresh K. Basandra,”Computers Today, 2005”, Galgotia Publications.

Semester - II

Semester-II (26 Credit Semester with 33 hours per week)

Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA551C	Data Structures	Core	5	3	0	4	07	FC
MCA552C	Artificial Intelligence	Core	5	3	0	4	07	AEC
MCA553C	Software Engineering	Core	4	3	1	0	04	FC
MCA554C	Operating System	Core	4	3	1	0	04	FC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MCA561E	Advanced Computer Organization & Architecture	DCE	3	2	1	0	03	AEC
MCA562E	Operational Research	DCE	3	2	1	0	03	AEC
MCA563E	Microprocessor & Assembly Language Programming	DCE	3	2	1	0	03	AEC
MCA564E	Soft Computing	DCE	3	2	1	0	03	SEC
Pool B: 03 Discipline Centric Elective Credit Units								
MCA576E	Modeling & Simulation	DCE	3	2	0	2	04	SEC
MCA577E	Computer Graphics	DCE	3	2	0	2	04	SEC
MCA578E	Numerical and Statistical Computing	DCE	3	2	0	2	04	VAC
MCA579E	Programming in Python	DCE	3	2	0	2	04	SEC
MCA580E	Java Programming	DCE	3	2	0	2	04	EC
Audit Courses (Mandatory)								
MCA591A	Technical Communication	Audit	0	2	0	0	02	VAC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			**26**				**33**	

Course Title:	Data Structures	**Course Code:**	MCA551C
Semester:	2nd	**Paper Type:**	Core
Credits:	05	**Max Marks::**	125
Pre Requisite:	MCA501C, MCA503C	**Co-Requisite:**	MCA552C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVE:

- The objective of this course is to introduce the students to the significant topics of basic as well as advanced data structures and subsequently make them understand & learn the commonly used data structures along with their applications. The focus is also to appreciate the need and working of different ways of storing data and demonstrate the advantages and disadvantages of specific data structures both linear as well as non-linear.
- To emphasizes on lab work wherein the students learn not only to make use of different data structures, but also their application in different synthetic problems. The practical demonstration for autonomous realization of simple programs or program parts is to understand the behavior of these basic data structures and to have a thorough understanding of how data structures influence the performance of algorithms.

COURSE CONTENT:

UNIT I

Data Structures: Overview & Significance. Introduction & Analysis of Algorithms. Linear & Non Linear Data Structures. Array & Strings, Application of Arrays, Sparse Matrix, Searching & Sorting: Linear & Binary Search, Bubble Sort, Selection Sort & Insertion Sort. Recursion & Applications.

UNIT II

Stack: Operations & Applications, Polish Expressions Queue: Operations & Applications, Circular Queue: Operations & Applications. Linked List: Singly Linked List, Doubly Linked List, Circular Linked List. Linked implementation of Stack & Queue, Operations & Applications of Linked Lists.

UNIT III

Trees: Basic Concept, Terminology & Representation of a Tree. Binary Tree, Binary Search Tree, Tree Traversal Techniques, BST Sort, AVL Trees, Heap Sort, Quick Sort, Merge Sort. Threaded Binary Tree, Red Black Trees, B Trees, B* Trees, Applications of Trees.

UNIT IV

Graph: Matrix & List Representation, Elementary Graph Operations (BFS & DFS), Single Source Shortest Path, Dijkstras Algorithm, Bellman-Ford Algorithm. All Pair Shortest Paths, Transitive Closure, Floyd-Warshall Algorithm. Spanning Trees: Minimum Cost Spanning Trees, Prims & Kruskal Algorithm. Hashing: Overview, Hashing Functions, Collision Resolution Techniques.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Develop some simple applications, like a desk calculator using stacks.
- Understand advanced searching methods like B-tree, B+ tree, AVL/red-black trees.
- Learn to develop a basic file system.
- Use standard libraries for data structures.

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Ed.

References:

1. Data structures and Algorithms in C++ -- by Adam Drozdek (1994 2001).
2. Fundamentals of Data Structures in C -- by Horowitz, Sahni and Anderson-Freed (Silicon Press 2007).
3. Data Structure Using C and C++ -- by Y. Langsam, M. J. Augenstein and A. N. Tanenbaum (Pearson Education, 2nd Edition, 2015).

List of Practicals:

LAB JOURNAL_MCA551C	
S. No.	**TITLE**
1.	Create One dimensional array of Integers and then perform the following operations on it: i. Read the Elements in an Array ii. Display the Elements in Input Order iii. Display the Elements in Reverse Order iv. Search a Particular Element v. Replace a Particular Element vi. Insert a Missed Element vii. Sort the Array Elements in any Order viii. Insert a Missed Element (after sorting) ix. Delete a Particular Element x. Frequency of a Particular Element xi. Display the Distinct Elements xii. Frequency of Distinct Elements xiii. Display the Duplicate Elements xiv. Remove the Duplicate Elements xv. Display the sum of all Elements xvi. Get the Average of all Elements xvii. Display the Maximum Value Element xviii. Display the Minimum Value Element
2.	Create a Two dimensional array of Integers and then perform the following operations on it: i. Read the elements of the 2D array ii. Display the elements iii. Get Transpose of Matrix (2D Array) iv. Perform the addition & subtraction of two matrices(2D arrays) v. Perform the Multiplication of Two Matrices(2D arrays)
3.	Generate the information matrix from a Sparse Matrix and then regenerate the Original Matrix.
4.	Perform a Linear Search as well as the Binary Search on an array of Integers
5.	Perform Bubble Sort & Insertion Sort on an array of Integers
6.	Create a Stack and perform all the permissible operations of Stack
7.	Create a Simple Queue and perform the permissible operations of Simple Queue
8.	Create a Circular Queue and perform the permissible operations of Circular Queue
9.	Convert an Infix expression to its corresponding postfix expression and then evaluate it as an application of Stack.
10.	Get the Factorial of a number using Recursion.
11.	Create a Singly Linked List and perform the following operations on it:

	i. Display the Node Information in Input Order ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information (at the Beginning, at Specific Location or at the end) v. Sort the Node Information in any Order(Prefer Insertion Sort) vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node(after sorting)k viii. Delete a Particular Node (at the Beginning, at Specific Location or at the end) ix. Find Frequency of a Particular Node Information x. Reverse the List
12.	Create a Doubly Linked List and perform the following operations on it: i. Display the Node Information in Input Order ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information (at the Beginning, at Specific Location or at the end) v. Sort the Node Information in any Order(Prefer Insertion Sort) vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node (after sorting) viii. Delete a Particular Node (at the Beginning, at Specific Location or at the end) ix. Find Frequency of a Particular Node Information x. Display the Node Information in Reverse Input Order
13.	Create a Circular Linked List and perform the following operations on it: i. Display the Node Information ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information v. Sort the Node Information in any Order vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node (after sorting) viii. Delete a Particular Node ix. Find Frequency of a Particular Node Information x. Display the Node Information in Reverse Order
14.	Create a Linked Stack and perform all the permissible operations of Stack
15.	Create a Linked Queue and perform all the permissible operations of Queue
16.	Create a Binary Tree and use the traversal Techniques to display the nodes/vertices of Binary Tree?
17.	Create a Binary Search Tree (BST) and then perform the following operations on it: i. Visit the nodes of BST (InOrder, PreOrder&PostOrder) ii. Delete any particular leaf node iii. Delete a node with only One SubTree (Left or Right) iv. Delete a node with both SubTrees
18.	Perform Selection Sort, Quick Sort & Merge Sort on an array of Integers.
19.	Perform the following operations on an AVL Tree: i. Insert an element into a AVL tree. ii. Delete an element from a AVL tree. iii. Search for a key element in a AVL tree.
20.	Perform Binary Search Sort & Heap Sort on a set of integers in a heap.
21.	Demonstrate the Threaded Binary Tree and perform all the permissible operations on it.
22.	Demonstrate the Red Black Treeand perform all the permissible operations on it.
23.	Demonstrate the B Tree and perform all the permissible operations on it.
24.	Demonstrate the B* Tree and perform all the permissible operations on it.
25.	Create Adjacency Matrix and Adjacency List for a defined Graph

26.	Perform Depth First Search & Breadth First Search on a Graph
27.	Generate all pair Shortest Paths in a Graph
28.	Compute Shortest Path between any two vertices in a Graph
29.	Perform the following: i. Dijkstra Algorithm ii. Bellman-Ford Algorithm iii. Floyd Warshall Algorithm
30.	Perform the Following in a Spanning Tree: i. Prim's Algorithm ii. Kruskal's Algorithm

Course Title:	Artificial Intelligence	**Course Code:**	MCA552C
Semester:	2nd	**Paper Type:**	DCE
Credits:	05	**Max Marks:**	125
Pre Requisite:	MCA503C,MCA529E	**Co-Requisite:**	MCA551C, MCA564E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To present an overview of artificial intelligence (AI) principles and approaches.
- To develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning.
- To have understating of different search problems and their solution using various algorithms.
- To have basic understanding of machine learning strategies and their role in Artificial Neural Networks

COURSE CONTENT:

UNIT I

Introduction to Artificial Intelligence, Applications of Artificial Intelligence, and Intelligent agents: Agents and Environments, the nature of environments, structure of agents, Concept of Rationality. Introduction to First order logic, rules in FOL, Prepositional Logic.

UNIT II

Searching for solutions, uniformed search strategies. Search with partial information (Heuristic search) Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions. Local search Algorithms: Hill climbing, local beam search. Game Playing: Adversial search, Games, minimax, algorithm, optimal decisions in multiplayer games, Alpha-Beta pruning. Exploring Search techniques with implementation in python.

UNIT III

Introduction to Learning, concept of supervised, unsupervised and reinforcement learning.

Supervised Learning Algorithms- Classification, ANN, KNN, SVM, Naïve Bayes algorithm, Decision Tree, Linear Regression, Advantages of supervised Learning. Exploring classification Models with implementation in Python using scikit-learn and visualize the results.

UNIT IV:

Unsupervised Learning Algorithms: Clustering, k-means clustering, hierarchical clustering, distributional clustering, Reinforcement learning; Learning from heterogeneous, distributed, data and knowledge. Reinforcement Learning Algorithms: characteristics of Reinforcement Learning- Types of Reinforcement Learning models. Exploring Clustering algorithm with implementation in Python using scikit-learn and visualizing the resulting clusters on a dataset.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Design intelligent machines/systems which act rationally and take the right decision at the right time.
- Define an AI problem and find a solution for it.
- Represent Knowledge using various knowledge representation schemes.
- Understand the basic knowledge acquisition methods.
- Understand Artificial Neural Networks and its applications

Text Books:

1. Artificial Intelligence – A Modern Approach. Second Edition, Stuart Russel, Peter Norvig, PHI/Pearson Education.
2. Introduction to Machine Learning with Python, Andreas C. Mueller

References:

1. Introduction to Artificial Intelligence – RajendraAkerkar, PHI.
2. Artificial Intelligence, 3rd Edition, Patrick Henry Winston., Pearson Edition
3. Artificial Intelligence and Expert Systems – Patterson PHI
4. Expert Systems: Principles and Programming- Fourth Edn, Giarrantana/ Riley, Thomson

List of Practicals:

LAB JOURNAL_ MCA552C	
S. No.	**TITLE**
1.	Implement Uninformed algorithms on any given problem
2.	Implement Informed algorithms on any given problem
3.	**Classification Models** Create a decision tree from given dataset using ID3 algorithm
4.	Given a dataset which is linearly separable, classify the datapoints using SVM.
5.	Given a 2-class dataset which is linearly inseparable, demonstrate what needs to be done so that SVM can be used to classify it
6.	Demonstrate Neural Networks using different activation functions
7.	Implement Back-propagation Algorithm in ANN
8.	**Clustering Models** Cluster a given 2-dimensional dataset using k-means algorithm
9.	**Game Playing** Implement the minimax algorithm to handle multiplayer games using Alpha-Beta pruning.

Course Title:	Software Engineering	**Course Code:**	MCA553C
Semester:	2nd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA501C, MCA502C	**Co-Requisite:**	MCA551C, MCA502C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To familiarize students with the fundamentals of software engineering.
- To introduce students to the basic concepts and principles of structured software engineering.
- To make students understand the software management issues.
- To introduce the students to the basic concepts of OOSE.
- To study advanced concepts in software engineering like Reverse engineering, Reengineering, etc.

COURSE CONTENT:

UNIT I

Software Engineering: Definition & Evolution, its Role & Impact in Computer Science. Software Process, Characteristics of a SW Process, CMMI, TSP & PSP, Software Product, Characteristics of a Good Software Product, Software Process Models, Comparative Study & Applications. Basic concepts of Agile Process. Software Requirements Analysis (SRA): Requirements - Types, Steps Involved in SRA. SW Requirements Specification (SRS): Need & Characteristics for an SRS, Components of an SRS, Prototype for a Good SRS. Structured Analysis: DFD'S, Control Flow Diagrams, Data Dictionary, State Transition Diagrams, & Entity - Relationship Diagrams. Case Study: Developing a Complete SRS.

UNIT II

Software Design: Concepts & Principles, Design Considerations & Good Design. Characterization of Effective Modular Design (Functional Independence, Cohesion, Coupling). Design: Architectural Design, Procedural Design, Interface Design, & Data Design. SW Architecture Styles: (Dataflow, Call & Return Architectures, Independent Process Architectures, Virtual Machine Architectures). Concept of Verification & Validation. Goals of SW Testing, Testing Principles.

UNIT III

Approaches to the Design of Test Cases: Black Box & White Box Testing, Techniques used by these Approaches: Basis Path & Loop Testing, Graph Based Testing, Equivalence Partitioning, Cyclomatic Complexity, Documentation of Test Cases, Phases in Testing Activity: Unit,

Integration, Validation & System Tests. Software Project Management, Phases of Management, Project Planning & Control, Scheduling, Organization & Team Structures, Project Estimation Techniques – KLOC, FP & COCOMO, Risk Analysis & Management, , "Risk Exposure",Software Quality Assurance, Software Configuration Management.

UNIT IV

Technical Metrics for Software. Object Oriented Software Engineering: Object Oriented Paradigm, Concepts - Classes & Objects, Inheritance, Abstraction, Polymorphism, etc. OOA & OOD. Design Methodology – Dynamic Modeling, Functional Modeling. Advanced Concepts: Software Reuse, Re-engineering, Reverse Engineering, Restructuring, Client/Server Software Engineering, Computer Aided Software Engineering, Advances & Future Scope in Software Engineering.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Describe software engineering layered technology and process framework.
- Understand theories, models, and techniques that provide a basis for the software development life cycle.
- Understand software testing approaches including verification and validation, static analysis, reviews, inspections, and audits.
- Understand the role of project management including planning, scheduling, risk management, etc.
- Work as an individual and/or in team to develop and deliver quality software.

Text Books:

1. Pressman, Roger, “Software Engineering- A Practitioners Approach”, McGraw Hill

References:

1. Gheezi, Jazayeri Et Al, “Fundamentals Of Software Engineering”, PHI
2. Ian Sommerville, “Software Engineering”, Pearson Education
3. PankajJalote, “An Integrated Approach To Software Engineering”,Narosa
4. Peters & Pedrycz, “Software Engineering an Engineering Approach”, Wiley

Course Title:	Operating System	**Course Code:**	MCA554C
Semester:	2nd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA501C, MCA502C,MCA504C	**Co-Requisite:**	MCA551C, MCA553C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management I/O management and file management in OS
- To gain knowledge on Mutual exclusion algorithms, deadlock detection algorithms
- To know the components and management aspects of concurrency management
- To implement the concepts learnt through case studies of different operating systems.

COURSE CONTENT:

UNIT I

Overview of an Operating system, Functions of Operating System. Process and thread Management Concepts: Process Scheduling, Scheduling Criteria, Scheduling algorithms, Thread Scheduling. Process Synchronization, Semaphores, Critical Section and Monitors, Inter-process Communication, shared memory model and message passing. Deadlocks: Concept of Deadlock, Deadlock prevention, avoidance, detection and Deadlock recovery.

UNIT II

Memory Management: Linking, Loading, Memory Allocation, Design Issues & Problems, Fragmentation, Compaction, Memory Management Unit, Paging, Segmentation, Virtual Memory, Demand Paging. Page Replacement Algorithms. Allocation Algorithms, Thrashing.

UNIT III

File Management: File Structure, File Protection, File System Implementation, Directory Structure, Free Space Management, Allocation Methods, Efficiency & Protection. Disk Management: Disk Structure, Disk Scheduling Algorithm, Disk Management, Swap Space concept & Management, RAID Structure, Disk Performance issues.

UNIT IV

Multiprocessor Systems: Types of Multiprocessor Operating Systems, Functions & Requirements, Design & Implementation Issues. Distributed Operating System: Difference between Distributed & Centralized Operating System, Advantages & Disadvantages of Distributed Operating System,

Hardware & Software Concepts, Loosely Coupled Systems, Types of Distributed Operating System.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Master understanding of design issues associated with operating systems.
- Master various process management concepts including scheduling, synchronization, and deadlocks.
- Be familiar with various types of operating systems including UNIX.

Text Books:

1. Tanenbaum, A.S., “Modern Operating System”, PHI

2. Peterson, J.L. Abraham, Silberschatz, “Operating System Concepts”, Addison Wesley

References:

1. Dietel H.M. “An Introduction To Operating System”, Addison Wesley
2. Karnetkar, “UNIX Shell Programming”, BPB
3. W.Stallings, “Operating systems”
4. Dhamdhere, “An Operating System –Design & principles”
5. Madnick E, Donovan J, “Operating Systems”, TMH
6. Marko Gergent, ”Learning android”, O’rielly

Course Title:	Advanced Computer Organization & Architecture	**Course Code:**	MCA561E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA504C, MCA511E	**Co-Requisite:**	MCA551C, MCA554C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To provide an insight about Classification of parallel computers, advanced processor, pipelining, memory design & architecture and multi-processor architecture.
- To Understand concepts of parallel processing and design choices of implementing parallel execution within a single processor (pipeline, VLIW, and superscalar) and multiprocessor systems.
- To gain knowledge of the state of the art research topics on advanced computing systems.

COURSE CONTENT:

UNIT I

Computational Models: Introduction, Interpretation of the concept of a computational model, Relationship between the concepts of computational model, programming language & architecture , Basic Computational models , The Von , Neumann computational model ,Key concepts related to computational models .The concept of computer architecture : Evolution & interpretation of the concept of Computer Architecture at different levels of abstraction. The concept of computer architecture at multilevel hierarchical framework.

UNIT II

Introduction to Parallel Processing: Basic Concepts about program, process, thread, process & treads in languages, concurrent & parallel execution, concurrent & parallel programming languages, Types & levels of Parallelism, Classification of Parallel architectures, Basic Parallel Techniques

UNIT III

Introduction to Instruction level Parallel Processors, Evolution & overview, dependencies, instruction scheduling, preserving sequential consistency, Pipelined Processors, Basic Concepts, Pipelined instruction processing. VLIW, Basic Principles, Superscalar processing, introduction, superscalar instruction issue, shelving, parallel execution, preserving the sequential consistency of instruction execution & exception processing

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the basic structure of computer.
- Familiarize the instructions in central processing unit of a computer.
- Understand memory organization in a computer.
- Understand input/output mechanisms.
- Understand parallel processing in a computer.

Text Books:

1. Advanced Computer Architecture DEZSO SIMA, TERENCE Mountain, PETER KACSUK, Pearson Education, Fifth Indian reprint 2004.

References:

1. V.C. Hamacher. A.G. Vranesic & S. G. Zaky: "Computer Organization", Tata McGraw Hill.
2. J.P. Hayes: "Computer Architecture & Organization", McGraw Hill.
3. Morris Mano: "Computer System Architecture", Pearson Education, 3/e.

Course Title:	Operational Research	**Course Code:**	MCA562E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA503C	**Co-Requisite:**	MCA551C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To identify and develop operational research models from the verbal description of the real system.
- To understand the mathematical tools that are needed to solve optimization problems.
- To define and formulate linear programming problems and appreciate their limitations.
- To understand how to solve linear programming problems using appropriate techniques and optimization solvers, interpret the results obtained and translate solutions into directives for action.
- To conduct and interpret post-optimal and sensitivity analysis and explain the primal-dual relationship.
- To develop mathematical skills to analyze and solve integer programming and network models arising from a wide range of applications.

COURSE CONTENT:

UNIT I

Linear Programming: L P formulation, Graphical methods for LPP with 2 variables, Simplex Algorithm for LPP, Duality theorem in linear programming & applications. Transportation problem: Formulation, methods of selecting initial feasible solutions, Degeneracy & resolution. Assignment problem: Hungarian Method for solving Assignment Problems, Balanced & Unbalanced problems & resolution.

UNIT II

Network Analysis: Shortest routes, Enumeration& applications. Critical Path Method (CPM) & Programme Evaluation and Review Technique(PERT): Use & design of CPM & PERT. Critical Path calculation, Float & its Types, Crashing in Project Management.

Sequencing problems, Johnsons algorithm for processing m jobs through 2, 3 & ‘n’ machines.

UNIT III

Replacement & Sequencing models: Replacement of items that fail & deteriorate. Group & individual replacement.

Game theory: definition & explanation, saddle points, Dominance mixed strategies, games without saddle points, 2*N games.

Dynamic Programming: Characteristics of dynamic programming problem, Bellman’s optimality principles, dynamic programming under certainty, shortest route problem,

COURSE OUTCOMES:

At the end of the course student will be able to:

- Formulate and solve linear Programming Problems
- Determine the optimum solution to constrained and unconstrained
- Apply dynamic programming principle to Linear programming problems.
- Determine the integer solutions to Linear Programming Problems

Text Books:

1. N.D.Vohra, “Quatitative Techniques in management”, TMH
2. Hamdy A. Taha, “Operations Research: An Introduction”, Pearson

References:

1. Sharma J. K., “Operations Research: Theory & Applications”, Macmillan India
2. Gross Donald, “Fundamentals of Queuing Theory”, 3rd Ed., John Wiley
3. Mokhtar S. Bazaraa, “Linear Programming & Network Flows”, John Wiley
4. Hiller Lieberman, “Introduction to Operations Research”, TMH
5. Laudon, “Decsion Support Systems”, PHI

Course Title:	Microprocessor & Assembly Language Programming	**Course Code:**	MCA563E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA504C, MCA511E,MCA512E	**Co-Requisite:**	MCA561E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand the principles of instruction set architecture and assembly language programming.
- To understand basic procedures of how a compiler translates C/C++ code to assembly language and perform simple optimizations.
- To understand basic principles of interrupt/exception handling.
- To explore in detail a simple hardware CPU implementation that supports a small instruction subset.

COURSE CONTENT:

UNIT I

Software Model of 8088 / 8086 Microprocessor, Memory Add. Space & Data Organization, Data Types, Segment Registers, Memory Segmentation Dedicated, reserved &general use of Memory, generating an Memory Address, Pin-out diagram of 8086 Microprocessor.

UNIT II

The Microcomputer Organization, Assembly Language Programming Development on PC, Instruction Set, Addressing Modes, 8086 Instruction set, Integer Instructions & Computations, Data Transfer, Arithmetic, Logic Shift, Rotate Instruction, Flag Control, Compare, Control Flow & Jump, Subroutine & Subroutine Handling Instructions, Loop & Loop Handling, String & String Handling Instructions. Statement Syntax for a source Program, Assembler Directives, Assembling, Linking, Loading & executing a run Module.

UNIT III

Isolated I / O, Memory Mapped I/O, DMA Controller, Programming Communication Interfaces Controller. Interfacing I/O devices to microprocessor, programmable peripheral interface. Interrupt, Mechanism, Types & Priority, Interrupt Vector table, Real Mode. 8086 / 8088 Microprocessors, 8086 / 8088 Microprocessor's Minimum Mode, Maximum Mode Systems, Bus Cycle & Unit States, Memory Control Signals, Read & Write Bus Cycles, Memory Interface Circuits.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Describe the architecture of 8086
- Develop simple program using 8086
- Describe the basic peripheral devices and its applications
- Differentiate various microprocessor architectures

Text Books:

1. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications (Prentice Hall international editions) Walter A. Triebel; Avtar Singh

References:

1. DOUGLAS HALL "Microprocessors & Interfacing" Tata McGrawHill.
2. LIU, GIBSON et al "Microcomputer system The 8086/8088 Family" PHI.
3. PAL CHAUDHURI "Computer Organization & Designing" PHI.
4. MORRIS MANO "Computer System Architecture" Pearson Education.
5. GILMORE "Microprocessors" Wiley/ Tata McGraw Hill.

Course Title:	Soft Computing	**Course Code:**	MCA564E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA551C, MCA503C	**Co-Requisite:**	MCA552C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To introduce soft computing concepts and techniques and foster their abilities in designing appropriate technique for a given scenario.
- To implement soft computing-based solutions for real-world problems.
- To give students knowledge of non-traditional technologies and fundamentals of artificial neural networks, fuzzy sets, fuzzy logic, genetic algorithms.

COURSE CONTENT:

UNIT I

Soft Computing & AI: Introduction & applications of soft computing, soft computing vs. hard computing, Artificial Intelligence: Introduction & application, Intelligent Agents, Structure of Intelligent Agents & types of agents. Introduction to Learning – Supervised, Unsupervised and Reinforcement Learning.

UNIT II

Neural Networks & Fuzzy Logic: Neural networks: Characteristics, artificial neural net terminology, Model of a neuron, McCullough-Pitts Model of a neuron, Single layer and Multilayer perceptron, sigmoid function, Training by back propagation, generalization, avoiding, over fitting.

UNIT III

Fuzzy set theory, Fuzzy set versus crisp set, Crisp relation & fuzzy relations, Fuzzy systems: crisp logic, fuzzy logic, introduction & features of membership functions. Introduction to Evolutionary computation: Evolutionary algorithms- Genetic & differential evolution, Swarm intelligence, Population based meta heuristic methods,

COURSE OUTCOMES:

At the end of the course student will be able to:

- Identify and describe soft computing techniques and their roles in building intelligent machines
- Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems and to apply genetic algorithms to combinatorial optimization problems.
- Evaluate and compare solutions by various soft computing approaches for a given problem.

Text Books:

1. Jyh:Shing Roger Jang, Chuen:Tsai Sun, EijiMizutani, Neuro: Fuzzy and Soft Computing, Prentice: Hall of India, 2003
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.

References:

1. Artificial Intelligence-A Modern Approach" – by Stuart Russell, Peter Norvig, , Pearson Education

2. Neural Networks, Fuzzy Logic & Genetic Algorithms, Synthesis & applications, S,Rajasekaranand G.A. VijayalakshmiPai, PHI Publication.
3. Introduction to artificial neural systems - by J.M. Zurada.(Jaico Pub)
4. Artificial Intelligence, Rich E & Knight K, TMH, New Delhi.
5. Neural Network fundamental with Graph , Algo.andAppl, by Bose, TMH
6. Neural Network & Fuzzy System, by Kosko PHI Publication
7. Neural Networks & Fuzzy Logic - by Bart Kosko
8. Neural computing theory & practice - by P.D. wasserman (ANZA PUB).
9. Principles of Soft Computing, S.N. Sivanandamand S.N. Deepa, Wiley Publications
10. Luger G.F. and Stubblefield W.A. (2008). Artificial Intelligence: Structures and strategies for Complex Problem Solving. Addison Wesley, 6th edition.

Course Title:	Modeling & Simulation	**Course Code:**	MCA576E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA503C	**Co-Requisite:**	MCA551C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand different methods for random number generation
- To understand various probability distributions
- To have a clear understanding of the need for the development process to initiate the real problem.
- To have a clear understanding of principle and techniques of simulation methods informed by research direction.
- To be able to describe the components of continuous and discrete systems and simulate them
- To be able to model any system from different fields
- To be able to implement numerical algorithm to meet simple requirements, expressed in English

COURSE CONTENT:

UNIT I

Concepts of Systems, Models, & Simulation. Distributed Lag Model, Cobweb Models, The process of a simulation Study, Exponential Growth Models, Exponential Decay Models, Type of simulation, Discrete-Event Simulation: Time-Advance Mechanisms, Components & Organization of a Discrete-Event Simulation Model. Monte Carlo Method. Simulation of Single-Server Queuing System, Simulation of an Inventory System

UNIT II

Continuous Simulation: Pure-pursuit Problem. Monte Carlo Method. Overview of Random numbers and Pseudo random numbers, Random Number Generators: Random Linear Congruently Generators, General Congruences, Composite Random number generator, Testing Random Number Generators. Generating Random Variates: General Approaches, Continuous & Discrete distributions.

UNIT III

Introduction to GPSS, GPSS block-diagrams, General Description, Simulation of a Manufacturing Shop. SNA, Function, Simulation of a Supermarket, GPSS Model of a Simple Telephone System

COURSE OUTCOMES:

At the end of the course student will be able to:

- Determine the properties of different types of physical systems and different types of simulations that are suitable to analyze their behaviors;
- Analyze data collected from real world and build input models for simulation studies;
- Conduct various simulation studies to investigate the behaviors of complex systems;
- Conduct statistical analysis of the simulation outputs; and
- Analyze discrete event systems through the competent use of computer simulation methods and mathematical modeling techniques.

Text Books:

1. Averill Law and Averill M. Law, Simulation Modeling and Analysis with Ex-pertÖt Software, Forth Edition, McGraw-Hill Higher Education, 2007

2. Jerry, Banks. Discrete event system simulation. Pearson Education India, 2005.
3. System Simulation – Geoffrey Gordon, 2nd Edition, PHI
4. System Simulation with Digital computer – Narsingh Deo, PHI

References:

1. Law & Kelton, “Simulation Modeling & Analysis”, McGraw Hill
2. Fred Maryanski, “Digital Computer Simulation”, CBSPD
3. James A. Pyne, “Introduction to Simulation- Programming Techniques & Methods of
4. Analysis”, McGraw Hill
5. Zeigler & Kim, “Theory of Modeling & Simulation”, Academic Press
6. Banks et al, “Discrete event Simulation”, Pearson Education

List of Practicals:

LAB JOURNAL_ MCA576E	
S. No.	**TITLE**
1.	WAP to Generate Random Number
2.	WAP to Simulate Random Version of Pure Pursuit Problem
3.	WAP to Simulate Actual Pure Pursuit Problem
4.	WAP to Simulate National Economy Model
5.	WAP to Simulate Exponential Growth Model
6.	WAP to Simulate Exponential Decay Model
7.	WAP to Simulate Cobweb Model
8.	WAP to Simulate Single Server Queuing Problem
9.	WAP to Simulate Inventory System
10.	WAP to Simulate Mean, Variance and Standard Deviation of Discrete Random Variable
11.	WAP to Simulate Mid Square Random Number Generator
12.	WAP to Simulate Linear Congruential Modulo Generator
13.	Multiplicative Random Number Generator
14.	Period Determination of Random Number Generator
15.	Quadratic Congruential Random Number Generator
16.	Fibonacci Random Number Generator
17.	Quadratic Random Number Generator
18.	Output Shuffle Random Number Generator
19.	Linear Feedback Shift Register
20.	Chi-Square Test
21.	Kolmogorov Smirnov Test
22.	Runs Up Runs Down Test
23.	Runs Above Runs Below Mean Test
24.	GPSS - Simulation of Supermarket
25.	GPSS - Simulation of Telephonic System

Course Title:	Computer Graphics	**Course Code:**	MCA577E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA503C	**Co-Requisite:**	MCA551C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To explain basic primitives like Point, Line, Circle, Ellipse etc. with their mathematical equations coupled with the geometrical interpretations.
- To explain process of displaying continuous primitives on Discrete Display Devices.
- To explain few transformations like Translation, Rotation, Scaling and composition of these transformations.
- To explain Projections from higher dimensional setup to lower dimensions.
- To explain drawing of complex scenes using Splines coupled with understanding of spline specifications and few variants of splines besides familiarizing about Beziers.
- To acquaint students about Concepts of displaying structures with no inherent regular geometry using Fractals.

COURSE CONTENT:

UNIT I

An Introduction Graphics System: Computer Graphics & Its Types, Application of computer graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Display Buffer, Concept of Double Buffering & Segmentation of Display Buffer. Use of Lookup tables. “Introduction to Color models (RGB, CMY, and HSV).”

UNIT II

Output Primitives & Attributes of Output Primitives: Output Primitives Points & Lines, Line Drawing Algorithms, Circle Generating Algorithms, Scan-Line Polygon Fill Algorithm, Inside-Outside tests, Boundary-Fill Algorithm, Flood Fill Algorithm, Cell Array, Character Generation, Attributes of Output Primitives: Line attributes, Color & Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes. Anti-aliasing.

Two-dimensional Geometric Transformations: Basic Transformations, Matrix Representations & Homogeneous Coordinates, Composite Transformations, Reflection & Shearing.

UNIT III

Two-Dimension Viewing: The viewing Pipeline, Window to view port coordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Text Clipping, Exterior Clipping.

Three–Dimensional Concepts: Three Dimensional Display Methods, 3D Transformations, Parallel Projection & Perspective Projection. Curves & Surfaces, Splines, Spline specification, Interpolated & Approximated Splines. Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. B-Splines curves & surfaces. Fractals - Fractal Generation Procedure.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand concepts of computer graphics.
- Understand 2D and 3D transformation, clipping, splines, objects modeling, colour modeling, lighting, textures, visible surface detection.
- Understand algorithms to design and create computer graphics scenes.

Text Books:

1. Hughes, John F., et al. Computer graphics: principles and practice. Pearson Education, 2014.

References:

1. W.M.Newman & Sproull. “Principles of interactive Computer Graphics” ,TMH
2. Steven Harrington.” Computer Graphics a Programming Approach” McGraw Hill.
3. Plastock & Kelley. “Schaums outline of theory & problems of computer Graphics”
4. David F Frogers & J Alan Adams. “Procedural Elements of Computer Graphics” McGraw Hill
5. David F Rogers & J Alan Adams. “Mathematical Elements of Computer Graphics” McGraw Hill

List of Practicals:

LAB JOURNAL_ MCA577E	
S. No.	**TITLE**
1.	Write a program in C/ C++ for displaying a line segment using the DDA algorithm.
2.	Write a program in C/ C++ for computing distance between two given points.
3.	Write a program in C/ C++ for finding midpoint between two given points in 2 dimensions
4.	Write a program in C/ C++ for displaying a line segment using Bressenham's Line drawing algorithms.
5.	Write a C++ to check whether a given line segment passes through a given set of points.
6.	Write a C++ program to find neighbors of a given point on circle with radius r and center (0, 0)
7.	Write a C++ program to find neighbors of a given point on circle with radius r and center (xc, yc)
8.	Write a program in C/ C++ to find whether a given point lies Inside, Outside or on the boundary of a given circle with center (xc, yc) and radius r.
9.	Write a program in C/ C++ to implement Mid-Point Circle Drawing Algorithm.
10.	Use polar coordinate system to disp[lay a circle with radius r and centre (xc, yc)
11.	Write a program in C/ C++ to implement a midpoint Ellipse drawing algorithm.
12.	Write a program in C/ C++ to do following basic 2-dimensional transformations i. Translation ii. Rotation iii. Scaling
13.	Write a program in C/ C++ to implement Basic/ Composite 2-dimensional transformations using matrix multiplication.
14.	Write a program in C/ C++ to draw a rectangular window of any arbitrary size with dimensions (Xwmin, Ywmin, Xwmax, Ywmax)
15.	Write a program in C/ C++ to draw a rectangular viewport window of any arbitrary size with dimensions (Xvmin, Yvmin, Xvmax, Yvmax)
16.	Write a program in C/ C++ to implement Window to Viewport transformation.
17.	Write a program in C/ C++ to implement flood fill algorithm
18.	Write a program in C/ C++ to fill polygon using flood-fill algorithm
19.	Write a program in C/ C++ to draw a rectangle
20.	Write a program in C/ C++ to drawing circle
21.	Write a program in C/ C++ to draw a circle without floating point arithmetic
22.	Write a program in C/ C++ to rotate an object by a given angle about a given point
23.	Write a program in C/ C++ for point clipping Algorithm
24.	Write a program in C/ C++ to implement Cohen Sutherland algorithm for line clipping.
25.	Write a program in C/ C++ for implementing Sutherland–Hodgman algorithm for polygon clipping

Course Title:	Numerical & Statistical Computing	**Course Code:**	MCA578E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA503C	**Co-Requisite:**	MCA552C, MCA562E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To explain concept of Polynomials, Equations, Zeros and Roots.
- To explain few iterative techniques for computation of roots together with their geometrical interpretation and implementation in C.
- To explain solving system of equation using known methods along with their geometrical interpretation and Implementation in c.
- To explain concepts of Curve Fitting with specific focus on Lagrange's interpolating Polynomial with a prior explanation of using linear and nonlinear curves for curve fitting.
- To explain few statistical concepts measures of central tendency, Dispersion etc.
- To explain concept of Hypothesis, Null hypothesis besides giving students knowledge about few tests which include T test, F test etc.
- To make students ready for courses like Artificial Intelligence and Machine Learning by giving them prior knowledge about Linear Algebra, Probability and statistics.

COURSE CONTENT:

UNIT I

Approximations & Errors – Types of Programming Errors, Data Errors, Computer & Arithmetic Errors, Round Off and Truncation Errors. Accuracy and Precision, Measures of Accuracy, Error Propagation. Non-Linear Equations, Types of Methods to find solutions to nonlinear equations, Algorithms to Compute Roots of Equation – Methods of Tabulation or Brute Force Method, Method of Bisection, Secant Method, Newton-Raphson Method, Method for False Position. Derivation of mathematical formulas, geometric interpretation and implementation of these methods.

UNIT II

Linear Equations, Types of Methods to find solutions to linear equations. Algorithms to Solve Linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Gauss Seidel, L.U. Decomposition, Lagrange Interpolated Polynomial, Newton's Methods of INTERPOLATION – Forward difference, Backward difference. Derivation of mathematical formulas and implementation of these methods.

UNIT III

Differential Equations – Concepts and Terminology, Algorithms to solve Ordinary Differential Equations – Euler Method and Modification. The trapezoidal Rule, Simpson's Rule. 4th order R-K Method.

Derivation of mathematical formulas and implementation of these methods.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the iterative methods to find solution of polynomial and transcendental equations.
- Discuss methods of interpolation and curve fitting.
- Find the solution of linear equations using matrices.

Text Books:

1. Rajaraman, Vaidyeswaran. Computer oriented numerical methods. PHI Learning Pvt. Ltd., 1993.

References:

1. S.C.ChapraandR.P.Canale: “Numerical methods for Engineering”. Tata McGraw Hill.
2. Krishenmurty & Sen : “Numerical Algorithms”
3. V. Rajaraman “Computer oriented numerical methods.” Prentice Hall of India.
4. McCalla, Thomas Richard: “Introduction to Numerical Methods & FORTRAN Programming”, John Wiley & Sons, Inc.
5. Grewal, B. S.: “Higher Engineering Mathematics”, Hindustan Offset Problems Series.
6. “SCHAUM’S Solved Problems Series”.
7. Sharma, K. D.:“Programming in Fortran”.
8. Jain, M. K., Iyengav, S. R. K., Jain, R. K.: “Numerical Methods for Scientific & Engineering Computation”+, Wiley Eastern Ltd, New Delhi.

List of Practicals:

LAB JOURNAL_ MCA578E	
S. No.	**TITLE**
1.	Write a program in C/ C++ to find real roots of nonlinear equation using Bisection Method
2.	Write a program in C/ C++ to find real roots of nonlinear equation using Regula Falsi or False Position Method
3.	Write a program in C/ C++ to find real roots of nonlinear equation using Newton Raphson Method
4.	Write a program in C/ C++ to find real roots of nonlinear equation using Secant Method
5.	Write a program in C/ C++ which solve systems of linear equation using Gauss Elimination Method
6.	Write a program in C/ C++ which solve systems of linear equation using Gauss Jordan Method
7.	Write a program in C/ C++ to find inverse of matrix using Gauss Jordan Method
8.	Write a program in C/ C++ to find largest or dominant Eigenvalue and corresponding Eigenvector using Power Method
9.	Write a program in C/ C++ to solve a system of linear equations using the Jacobi Iteration Method.
10.	Write a program in C/ C++ for solving systems of linear equations using Gauss Seidel iterative method
11.	Write a program in C/ C++ to generate forward difference table using discreetly defined Function
12.	Write a program in C/ C++ to generate backward difference table using discreetly defined function
13.	Write a program in C/ C++ which implements Lagrange Interpolation
14.	Write a program in C/ C++ which implements Linear Interpolation Method
15.	Write a program in C/ C++ which reads n data points from user and then implement linear regression on the given data.
16.	Write a program in C/ C++ which implements pseudocode for Curve Fitting of Type $y=aX^b$
17.	Write a program in C/ C++ which implements pseudocode for Curve Fitting of Type $y=ab^x$
18.	Write a program in C/ C++ to find derivatives using Newton's forward difference formula.
19.	Write a program in C/ C++ to find derivatives using Newton's backward difference formula.
20.	Write a program in C/ C++ to find numerical integration using Trapezoidal Rule or Method.
21.	Write a C program to implement R. K. 2nd order method for computing integral of a given function.
22.	Write a C program to implement R. K. 4th order method for computing integral of a given function.

Course Title:	Programming in Python	**Course Code:**	MCA579E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks::**	75
Pre Requisite:	MCA501C, MCA502C, MCA503C	**Co-Requisite:**	MCA551C

Marks Distribution: (Mid Term:30, End Term:35,Viva:5, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To learn the essential concepts of Python programming
- The course aims at equipping participants to be able to use python programming for solving data science problems.
- To perform data visualization and presentation of preprocessed data.
- To understand the advantage of using Python libraries for data analysis and implementing Machine Learning models

COURSE CONTENT:

UNIT I

Introduction to Python and its history, setting up Python development environment (IDEs, tools), Python syntax, indentation, and coding conventions.

Variables, identifiers, and reserved keywords; Numeric data types (int, float) and arithmetic operations; Conditional statements (if, elif, else), Looping constructs (for, while), Using break and continue in loops.

Python Data structures - lists, tuples, dictionaries, sets; List manipulation – indexing and slicing. Dictionary manipulation; String, list and dictionary in-built functions.

UNIT II

Introduction to function - Defining and calling functions, Function parameters, return values and scope, Lambda functions; Generators and Iterators. Introduction to regular expressions, Pattern matching and substitution. File Handling - Reading and writing files in Python, Working with CSV and JSON files. Object Oriented Programming – Introduction to OOPs concepts like classes, objects; inheritance and polymorphism.

UNIT III

Numpy: NdArray and its operation, Indexing, slicing, broadcasting. Boolean arrays, shape manipulations. Pandas -pandasdataframes, reading data from files, Data preparation and preprocessing. Data visualization on dataset using matplotlib/seaborn library: Scatter plot, Line plot, Bar plot, Histogram, etc. Introduction to machine learning (scikit-learn).

COURSE OUTCOMES:

At the end of the course student will be able to:

- Read, write, and execute simple Python programs.
- Write simple Python programs for solving problems.
- Decompose a Python program into functions, lists etc.
- Read and write data from/to files in Python Programs
- Perform Exploratory Data Analysis

Text Books:

1. Mastering Python for data science, Samir Madhavan

2. Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning,ISBN: 978-1111822705.
3. Introduction to Machine Learning with Python, Andreas C. Mueller

References:

1. Machine Learning using Python, U Dinesh Kumar Manaranjan Pradhan
2. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.
3. Jake VanderPlas "Python Data Science Handbook" O'Reilly Publications.
4. Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.".
5. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

List of Practicals:

LAB JOURNAL_MCA579E	
S. No.	**TITLE**
1.	Write a program to demonstrate different number of data types in python.
2.	Write a program to perform different arithmetic operations on numbers in python.
3.	Write a program to create, concatenate and print a string and accessing substring from a given string.
4.	Write a python script to print the current date in following format "Sun May 29 02:26:23 IST 2017"
5.	Write a python program to create, append and remove lists in python.
6.	Write a program to demonstrate working with tuples in python
7.	Write a program to demonstrate working with dictionaries in python
8.	Write a python program to find largest of three numbers
9.	Write a python program to convert temperature to and from Celsius to Fahrenheit.
10.	Write a python program to construct the following pattern using nested for loop:
11.	Write a python program to print prim numbers less than 20:
12.	Write a python program to find factorial of a number using recursion:
13.	Write a python program to that accepts length of three sides of a triangle as inputs. The program should indicate whether or not the triangle is a right-angled triangle (use Pythagorean theorem):
14.	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15.	Write a python program to define a module and import a specific function in that module to another program.
16.	Write a program to demonstrate the use of various operators in python.
17.	WAP to demonstrate the implementation of various control flow structures in python.
18.	Write a Program to demonstrate the use of various looping structures in python.
19.	Write a Program to demonstrate the implementation of functions, modules and packages in python
20.	Write a Program to demonstrate various operations associated with files(Creating, opening, reading, writing, delete, update)
21.	Write a Program to demonstrate the implementation of Lists and its associated operations.
22.	Write a Program to demonstrate the implementation of Tuples and its associated operations.
23.	Write a Program to demonstrate the implementation of Dictionary and its associated operations.
24.	Write a Program to demonstrate the implementation of Strings and its associated operations.
25.	Write a Program to demonstrate the implementation of Sets and its associated operations.

26.	Write a Program to demonstrate the use of Numpy to perform various operations (Creating an Array (1-D,2-D,3-D), Adding, subtracting, Multiplying, Indexing, Slicing etc).
27.	Using a numpy module create an array and check the following: i. Type of array ii. Axes of array iii. Shape of array iv. Type of elements in array
28.	Using a Numpy module create array and check the following: i. List with type float ii. 3*4 array with all zeros
29.	Write a program to demonstrate the implementation of various Linear Algebra and related operations using python.
30.	Write a program to demonstrate the use of Pandas Library in Python
31.	Write a Pandas program to create and display a one-dimensional array-like object containing an array of data using Pandas module
32.	Write a pandas program to convert a pandas module series to python list and its type
33.	Write a pandas program to add, subtract, multiple and divide two pandas series
34.	Write a pandas program to convert a Numpy array to a pandas series.
35.	Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.
36.	Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame.
37.	Write a Pandas program to insert a new column in existing DataFrame
38.	Write a Pandas program to get list from DataFrame column headers.
39.	Write a Pandas program to join the two given dataframes along rows and assign all data.
40.	Write a Pandas program to read excel and CSV file to perform operations: drop null values, locate null values, fill null values, display file contents, etc.
41.	Write a python program to demonstrate Data visualization on dataset using matplotlib and seaborn libraries: Scatter plot, Line plot, Bar plot, Histogram, Box plot, Pair plot.

Course Title:	Java Programming	**Course Code:**	MCA580E
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA512E,,MCA526E,	**Co-Requisite:**	MCA552C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To cover OOPS concepts like inheritance, polymorphism, overriding etc.
- To introduce Lambda expressions, inner classes and interfaces.
- To learn about threads, assertions, logging and exception handling.
- To know about collection framework and introduction of spring framework.

COURSE CONTENT:

UNIT I

Fundamental Programming structures in Java: A simple java program, comments, data type, variables, operators, string, input& output, control flow, big numbers, arrays. Objects & Classes: introduction to Object Oriented programming, using predefined classes, defining your own classes, static fields & methods, method parameters, object construction, packages, the class path, documentation comments.

UNIT II

Inheritance Classes superclasses, subclasses,object:the cosmic superclass, generic array list, object wrapping & autoboxing, method with a variable number of parameters, enumeration class, reflection, design hints for inheritance. Interfaces lambda expressions & inner classes. Interfaces, examples of interfaces, lambda expressions, inner classes, proxies. Exceptions, assertion & logging dealing with errors, catching exception, using assertion, logging.

UNIT III

Generic Programming: why generic programming, defining a simple generic class, generic methods, bound for type variables, generic code & virtual machine restriction & limitation, inheritance rules for generic types, wildcard types, reflection & generics. Collections Java collection framework, concrete frameworks, Maps, Views & wrappers,Algorithms,Legacy collections.

Concurrency, Introduction to threads, Interrupting threads, thread states, thread properties, synchronization, blocking queues, thread safe collections, callable & futures, executors, synchronizers

COURSE OUTCOMES:

At the end of the course student will be able to:

- Describe the features of Java
- Design classes with object-oriented features
- Describe advanced features of Java like exception handling, multi-threading etc.
- Write programs in JAVA featuring its core capabilities

Text Books:

1. Core Java Volume 1-Fundamentals by Cay S.horstman 10th edition Publisher: Prentice Hall

References:

1. Java the complete reference by Herbert Schield 10th edition Publisher:Tata Mc Graw Hill
2. Java How to program Dietel and Dietel

List of Practicals:

LAB JOURNAL_ MCA580E	
S. No.	**TITLE**
1.	Write a program to implement different primitive data types.
2.	Write a program to implement different types of operators in Java.
3. L	Write a program to develop a simple cipher in Java using bitwise operators.
4.	Write a program to input two integers from the keyboard and display their sum.
5.	Write a program to implement a class "Test" in Java that defines a static field and a static method to initialize the field. Use another class to create an instance of the class "Test" and access its members using the class name and instance name.
6.	Create a package "MyClasses" and add the class "Test" to the package. Use another class to import the MyClasses package to use the Test class by calling its members.
7.	Write a program to swap two values using object reference. Your program should have a swap function.
8.	Write a class Employee and a class Teacher. Add general properties and methods of employees in the Employee class and add the special properties and methods of teacher in the Teacher class.
9.	Write a Java program that creates a 2D integer array with 5 rows and varying number of columns in each row. Using 'for each' variant of for loop display each element of every row.
10.	Write a Java program that calculates factorial of a number recursively.
11.	WAP that describes a class person. It should have instance variables to record name, age and salary. Create a person object. Set and display its instance variables.
12.	WAP that implements method overloading.
13.	WAP that shows passing object as parameter.
14.	WAP that illustrates method overriding
15.	Write a program to show that the value of non static variable is not visible to all the instances, and therefore cannot be used to count the number of instances.
16.	WAP to illustrate simple inheritance
17.	WAP illustrating a super class variable a referencing as sub class object.
18.	WAP illustrating all uses of super keywords.
19.	Create an abstract class shape. Let rectangle and triangle inherit this shape class. Add necessary functions.
20.	Write a Java program in which a Class overloads a method sum(), which takes 2 parameters. The overloaded methods should perform summation of either integer or floating-point values.
21.	Write a program to print the public fields, constructors, methods their return types and argument of a class using reflection.
22.	Write a java package to show dynamic polymorphism and interfaces.
23.	Write a program to implement a lambda expression with a suitable functional interface that takes two integers and return their sum.
24.	Write at least three interfaces and write at least one default method in any one of the interfaces. Write a class and implement all the interfaces in a class.
25.	Write an application that shows the usage of try, catch, throws and finally.
26.	Write an application that shows how to create a user-defined exception.

27.	Write a class with a method that throws an Arithmetic Exception. Use try catch blocks in the code calling the method to handle the exception.
28.	Write a program to implement a generic class and a generic method that receive different types of data and prints them.
29.	Implement different legacy collections in Java.
30.	Implement stack and queue in Java
31.	Write a program to use Maps collection in Java.
32.	Write a program to demonstrate the creation and usage of multiple threads in Java.
33.	Write an application that executes two threads. One thread displays —An‖ every 1000 milliseconds and other displays —B‖ every 3000 milliseconds. Create the threads by extending the Thread class.
34.	Write a multithreaded program that demonstrates the problem of race conditions.
35.	Write a program to demonstrate different concurrency control mechanism to avoid race conditions in multithreaded java programs.

Course Title:	Technical Communication	**Course Code:**	MCA591A
Semester:	2nd	**Paper Type:**	Audit
Credits:	00	**Max Marks:**	50
Pre Requisite:	--------	**Co-Requisite:**	-------------

Marks Distribution: (Mid Term:15, End Term:25, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The objective of studying Technical Communication course is that it enhances the employability skills and performance at workplace. It provides discussion on modern media tools for enriching presentation skills for preparing PowerPoint slides. Thus acts as a reference for training programs offered by business houses and Industries.

COURSE CONTENT:

UNIT I

Technical Communication: Basics of Technical Communication, Barriers to communication and Technology in Communication. Communication in the Workplace: Problem Solving in Workplace Communication, Human factors in the communication failure. Guidelines for ethical communication. Active Listening: Introduction, types of listening, traits of a good listener, Active vs. Passive listening and implications of good listening.

UNIT II

Introduction to Effective Presentation strategies: defining purpose, analysing audience and locale, organizing contents, preparing outline, visual aids, understanding nuances of delivery, kinesics, proxemics, paralinguistic, chromatics. Interviews: introduction, objectives, types of interviews and job interviews, guidelines for surviving a job interview. Group Communication: Introduction, Group discussion, Organizational Group discussions and meeting conferences. Technical Proposals: Definition, purposes, Types, Characteristics, elements of structure, style and Appearance, evaluation.

COURSE OUTCOMES:

At the end of the course, students will demonstrate proficiency by:

- Understanding the characteristics of technical writing and the importance of purpose, audience, and genre for written communication in technical fields.
- Articulating complex engineering ideas appropriate for targeted audiences.
- Planning, drafting, revising, editing, and critiquing technical and professional documents through individual and collaborative writing.
- Writing effective technical and business documents that are grammatically and stylistically correct.
- Preparing and delivering professional technical presentations through applying principles of effective oral communication and slide design.
- Applying principles for the visual display of quantitative information.
- Researching, analyzing, synthesizing, and applying information to create technical reports.
- Recognizing ethical implications of technical communication in professional contexts.
- Understanding the contemporary issues in engineering from an environmental, societal, economic, and global perspective.

Text Books:

1. William Pfeiffer, Padmaja,"Technical Communication A Practical Approach", Pearson Education.

References:

1. Meenakshi Raman & Sangeeta Sharma,"Technical Communication",Oxford University Press

Semester - III

Semester-III (26 Credit Semester with 31 hours per week)

Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA601C	Full-Stack Development	Core	5	3	0	4	07	EC
MCA602C	Machine Learning	Core	5	3	0	4	07	AEC
MCA603C	Data Communication & Computer Networks	Core	4	3	1	0	04	FC
MCA604C	Analysis and Design of Algorithms	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MCA611E	Advanced Operating Systems	DCE	3	2	1	0	03	AEC
MCA612E	Distributed Databases	DCE	3	2	1	0	03	SEC
MCA613E	Bioinformatics	DCE	3	2	1	0	03	AEC
MCA614E	System Software Design	DCE	3	2	1	0	03	AEC
MCA615E	Natural Language Processing	DCE	3	2	1	0	03	AEC
Pool B: 3 Discipline Centric Elective Credit Units								
MCA626E	Digital Image Processing	DCE	3	2	0	2	04	AEC
MCA627E	Linux System Programming	DCE	3	2	0	2	04	SEC
MCA628E	Data Warehousing	DCE	3	2	0	2	04	EC
MCA629E	Open Source Technologies	DCE	3	2	0	2	04	EOC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			**26**				**31**	

Course Title:	Full-Stack Development	**Course Code:**	MCA601C
Semester:	3rd	**Paper Type:**	Core
Credits:	05	**Max Marks::**	125
Pre Requisite:	MCA501C, MCA580E	**Co-Requisite:**	MCA629E

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- Understand the fundamentals of ASP.NET Core, MVC architecture, and JavaScript for full-stack web development.
- Gain proficiency in creating responsive and dynamic web applications using ASP.NET Core MVC and JavaScript.
- Develop skills in backend development, including database connectivity, authentication, and RESTful API implementation.
- Learn client-side scripting techniques with JavaScript and jQuery for enhancing user interactivity.
- Acquire hands-on experience in deploying and maintaining ASP.NET Core MVC applications on hosting platforms like Azure/ AWS/ GCP.

COURSE CONTENT

UNIT I

Overview of .NET Framework and .NET Core, Basic syntax and structure of C# programs, Data types, variables, and constants, Conditional statements (if, else, switch), Looping constructs (for, while, do-while, foreach), Exception handling (try, catch, finally).

Classes and objects, Properties, methods, and fields, Constructors and destructors, Inheritance and polymorphism, Basics of Reading from and writing to files

UNIT II

Introduction to Model-View-Controller (MVC) pattern, Creating models, views, and controllers, Routing and URL mapping, Razor syntax and view engine, Creating and processing forms, Model binding and validation, Tag Helpers and HTML Helpers, Partial views and view components.

Understanding dependency injection (DI) in ASP.NET Core, Configuring and using DI.

Introduction to ASP.NET Core Identity, Implementing authentication and authorization.

UNIT III

Overview of Entity Framework Core (EF Core), Setting up EF Core in an ASP.NET Core project, Creating and configuring the DbContext, Defining and mapping entities, Creating, reading, updating, and deleting (CRUD) operations

LINQ queries and lambda expressions, Database migrations and schema updates, Relationships (one-to-one, one-to-many, many-to-many), Building basic ASP.NET Core application with EF Core

UNIT IV

Frontend Development with JavaScript and jQuery: Introduction to client-side scripting with JavaScript (ES6+ Concepts), DOM manipulation and event handling, JavaScript functions and closures.

Introduction to jQuery library and its utility in web development, Integrating JavaScript and jQuery with ASP.NET Core MVC applications, Implementing client-side form validation and interactivity.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Demonstrate proficiency in setting up development environments for ASP.NET Core MVC and JavaScript projects and to Design & implement MVC architecture for building scalable and maintainable web applications.
- Utilize Entity Framework Core for database operations, implement validation and error handling mechanisms, also implement authentication and authorization features using ASP.NET Identity for secure user management.
- Develop interactive user interfaces using JavaScript and jQuery for dynamic content manipulation and event handling.
- Integrate frontend and backend components seamlessly through AJAX requests and API consumption.
- Construct single-page applications (SPAs) using ASP.NET Core MVC and JavaScript frameworks for enhanced user experience.
- Employ testing and debugging techniques to ensure the functionality and reliability of full-stack applications.
- Deploy ASP.NET Core MVC applications to cloud hosting platforms and manage application lifecycles effectively.

Text Books:

1. Freeman, Adam. "Pro ASP.NET Core MVC 2." Apress, 2017.
2. Duckett, Jon. "JavaScript and JQuery: Interactive Front-End Web Development." Wiley, 2014.

References:

1. Lock, Andrew. "ASP.NET Core in Action." Manning Publications, 2018.
2. Esposito, Dino. "Programming ASP.NET Core." Microsoft Press, 2018.
3. Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media, 2020.
4. Freeman, Adam. "Pro jQuery in ASP.NET Core MVC." Apress, 2018.
5. Schwartz, Carl. "A Practical Guide to ASP.NET Web API." Apress, 2013.
6. Wilson, James, et al. "Professional ASP.NET MVC 5." Wrox, 2014.

List of Practicals:

LAB JOURNAL_ MCA601C	
Week No.	**TITLE**
1 to 4	• Set up development environment with Visual Studio and create a basic ASP.NET Core MVC project. • Implement models, views, and controllers for a simple CRUD (Create, Read, Update, Delete) application. • Configure routing and request handling in ASP.NET Core MVC. • Enhance the CRUD application by adding form validation and error handling mechanisms.
5 to 8	• Integrate Entity Framework Core for database connectivity and implement data models. • Implement authentication and authorization using ASP.NET Identity. • Extend the application to include RESTful APIs for data manipulation. • Deploy the application to a local development server or cloud platform like Azure.
9 to 12	• Integrate JavaScript for client-side interactivity and basic DOM manipulation. •

	• Implement form validation using JavaScript. • Explore advanced JavaScript concepts like asynchronous programming and closures. • Integrate jQuery library into the ASP.NET Core MVC project for enhanced DOM manipulation and event handling.
13 To 16	• Implement AJAX requests to retrieve and update data asynchronously from the server. • Build a single-page application (SPA) using JavaScript frameworks like React, Angular, or Vue.js with ASP.NET Core MVC backend. • Final project: Students develop a full-stack web application that incorporates all concepts learned throughout the course, including backend connectivity, client-side interactivity, and real-time communication.

Course Title:	Machine Learning	**Course Code:**	MCA602C
Semester:	3rd	**Paper Type:**	Core
Credits:	05	**Max Marks::**	125
Pre Requisite:	MCA552C, MCA529E, MCA579E	**Co-Requisite:**	MCA604C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To explain Curve fitting with a specific focus on Regression analysis, logistic Regression Analysis.
- To explain different machine learning procedures which include Supervised, Unsupervised, semi supervised and reinforcement learning etc.
- To explain familiar machine learning Techniques like Support Vector Machines and its nonlinear variant.
- To explain few distribution like Gaussian, Dirichlet etc. and its subsequent use in modelling data for learning.
- To explain common applications of machine learning with a specific focus on latest developments in the said area.

COURSE CONTENT:

UNIT I

Foundations: Definition of learning systems, learning types (supervised, unsupervised, reinforcement).Goals and applications of machine learning in various domains.

Machine learning process overview: data acquisition, cleaning, pre-processing, feature engineering, model selection, training, evaluation, deployment.

Concept Learning: Hypothesis space and inductive bias. General-to-specific ordering of hypotheses and greedy search algorithms. Importance of simplicity and overfitting avoidance. Introduction to model evaluation metrics (accuracy, precision, recall, F1-score).

UNIT II

Decision Trees: ID3 algorithm, entropy-based attribute selection. Overfitting, pruning techniques, and their impact. Ensemble methods: Bagging, Boosting.

Linear Regression: Linear, multiple.

Bayesian Learning: Probability theory basics (conditional probability, Bayes' rule). Naive Bayes classifier and parameter smoothing. Logistic regression for classification.

Support Vector Machines (SVMs): Maximum margin hyperplane concept and kernel methods. Applications of SVMs in different domains.

UNIT III

Artificial Neural Networks (ANNs): Biological inspiration, perceptron's and limitations. Multilayer networks, backpropagation algorithm, and gradient descent. Hidden layers and distributed representations. Overfitting prevention techniques (regularization, dropout, cross-validation). **Unsupervised Learning**: Clustering - Hierarchical clustering and k-means clustering. Applications of clustering in data exploration and segmentation.

Instance-Based Learning: k-Nearest Neighbor (k-NN) algorithm and its variations. Applications of k-NN in recommendation systems and anomaly detection.

Deep Learning (Introduction): Brief overview of deep learning architectures (Convolutional neural networks, recurrent neural networks).Applications of deep learning in computer vision, natural language processing, and other fields.

UNIT IV

Deep Learning Frameworks: Introduction to PyTorch, its basic functionalities and syntax. Building and training simple deep learning models for classification and regression.

Transformers and Hugging Face: Introduction to Transformers and their applications in NLP tasks. Using pre-trained models from Hugging Face for text classification, summarization, and question answering.

Capstone Project:

Students choose a project in either NLP or computer vision domain.

Projects should utilize PyTorch and/or Hugging Face libraries.

Project deliverables include code, documentation, and presentation.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Apply a variety of learning algorithms to data.
- Perform evaluation of learning algorithms and model selection?
- Apply machine learning algorithms to solve problems of moderate complexity

Text Books:

1. Michalski, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. Machine learning: An artificial intelligence approach. Springer Science & Business Media, 2013.

References:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin

List of Practicals:

LAB JOURNAL_ MCA602C	
S. No.	**TITLE**
1.	NumPy i. Write a program in python to create a Numpy array filled with all zeros ii. Write a program in python to create a Numpy array filled with all ones iii. Write a program in python to flatten a Matrix using NumPy iv. Write a program in python to count the number of non-zero values in the array v. Write a program in python to count the number of elements along a given axis in Numpy array vi. Write a program in python to trim the leading and/or trailing zeros from a 1-D array vii. Write a program in python to change data type of given numpy array viii. Write a program in python to reverse a numpy array ix. Write a program in python to make a NumPy array read-only
2.	Pandas i. Write a program in python to make a Pandas DataFrame with two-dimensional list ii. Write a program in python to create DataFrame from dict of lists

	iii. Write a program in python to convert list of nested dictionary into Pandas dataframe iv. Write a program in python to replace values in Pandas dataframe using regex v. Write a program in python which demonstrates different ways to iterate over rows in Pandas Dataframe vi. Write a program in python for ranking Rows of Pandas DataFrame vii. Write a program in python for Sorting rows in pandas DataFrame
3.	Matplotlib i. Write a program in python to demonstrate how to add Title to Subplots in Matplotlib? ii. Write a program in python to demonstrate how to Set a Single Main Title for All the Subplots in Matplotlib? iii. Write a program in python to Increase the thickness of a line with Matplotlib iv. Write a program in python to Fill Between Multiple Lines in Matplotlib? v. Write a program in python to create a Scatter Plot with several colors in Matplotlib? vi. Write a program in python to increase the size of scatter points in Matplotlib ?
4.	Machine Learning i. Write a python script to implement Linear Regression using sklearn ii. Write a python script to implement Linear Regression Using Tensorflow iii. Write a python script for Boston Housing Kaggle Challenge with Linear Regression iv. Write a python script to implement Polynomial Regression from Scratch v. Write a python script to implement Logistic Regression using SKLearn vi. Write a python script to implement Logistic Regression using Tensorflow vii. Write a python script to implement Naive Bayes using SKlearn viii. Write a python script to implement Naive Bayes from scratch ix. Write a python script to implement Support Vector Machine Algorithm using SKLearn x. Write a python script for tuning SVM Hyperparameter using GridSearchCV xi. Write a python script to implement K-means Algorithm using SKLearn
5.	Deep Learning i. Write a python script to implement ANN from scratch ii. Write a python script to implement artificial neural network with Keras iii. Write a python script to implement Dog-Cat classifier using keras API iv. Write a python script which demonstrates how to Save and load a Keras model v. Write a python script to Implement Transfer Learning in Keras vi. Write a python script to Implement Dog-Cat Classifier using PyTorch

Course Title:	Data Communication & Computer Networks	**Course Code:**	MCA603C
Semester:	3rd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA511E	**Co-Requisite:**	MCA629E

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand networking concepts and basic communication model.
- To understand network architectures and components required for data communication.
- To analyze the function and design strategy of physical, data link, network layer and transport layer
- To acquire basic knowledge of various application protocol for internet security issues and services.

COURSE CONTENT

UNIT I

Principles of Data Communication, Data Communication Model & Tasks, Concept of Bandwidth & Channel Capacity, Nyquist's Law& Shannon's Law for Quantification of Capacity, Data Rate Versus Baud Rate, Sampling & its types, Nyquist Criterion for Sampling, Data Transmission Concepts, Characteristics of Signals: Digital Signals, Analog Signals (Amplitude, Frequency, Period, Wavelength, Signal to Noise Ratio).

UNIT II

Data Encoding: NRZ-L, NRZ-I Encoding, Multilevel Binary & Biphase Encoding Techniques & their implementations. ASK, FSK, PSK & QPSK. PCM: Sampling & Quantization & Modulation, Delta Modulation, Amplitude Modulation. Reliable Transmission of Data: Asynchronous & Synchronous Transmission, Multiplexing: Concepts & Types (FDM, Synchronous & Statistical TDM). Error Detection: Parity Based, CRC-Based, Checksum Based. Error Control & Recovery Techniques, Concept of ARQ Standard & its Versions.

UNIT III

Goals & Applications of Computer Network. OSI model. TCP/IP protocol suite. LAN: Ethernet & Token Ring. WAN: Concept, Characteristics, & Architecture: WAN subnet. Internetworking Concept & Architectural Model, Connection Oriented & Connectionless Approaches, Packet switching & Circuit switching. Concept of Autonomous Systems. Internet Layer Protocols: IP (Addressing: Classful & classless IP Addressing, IP Multicasting, Routing, Fragmentation& Reassembly).

UNIT IV

ICMP, ARP, RARP. Routing Protocols: Interior (OSPF), Exterior (BGP). Transport Layer Protocols: TCP & UDP. Socket API for Network Programming: Concept of Port & Sockets. Basic Server Architectures. TCP Server & Client. UDP Server & Client. Network Security: Principles of Security, Firewalls & their Components, Encryption Techniques & examples of Encryption Standard (DES, AES & RSA).

COURSE OUTCOMES:

At the end of the course student will be able to:

- Describe how to connect machines in a network.
- Describe data communication between machines at various locations.
- Familiarize with LAN connecting devices and Network Layer.
- Learn different Network Layer Protocols.
- Understand the concept of Domain Name System (DNS).

Text Books:

1. Computer Networks ,A.S. Tanenbaum, Pearson Education
2. Data Communications & Networking ,Forouzan, Tata McGraw Hill Company

References:

1. Computer Network, S.S.Shinde, New Age International Publisher.
2. Data & Computer Communication , Shashi Banzal ,Firewall media
3. Data & Computer communication, William Stallings, Pearson

Course Title:	Analysis & Design of Algorithms	**Course Code:**	MCA604C
Semester:	3rd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA501C, MCA551C, MCA552C	**Co-Requisite:**	MCA602C, MCA616E

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVE:

- The objective of the course is to teach techniques for effective problem solving in computing. The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problem. In each case, emphasis will be on rigorously proving correctness of the algorithm. In addition, the analysis of the algorithm will be used to show the efficiency of the algorithm over the naive techniques.

COURSE CONTENT:

UNIT I

Algorithms: Introduction, Importance, Performance Analysis, Designing Algorithms, Growth of Functions, Asymptotic Notations, Solving Recurrences. Review of Data Structures: Stacks, Queues, Trees, Graphs & Hashing.

UNIT II

Divide & Conquer: General Method, Binary Search, MinMax Problem, Merge Sort, Quick Sort, Strassen's Matrix Multiplication. Greedy Method: General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum Cost Spanning Trees, Optimal Storage on Tapes, Optimal Merge Pattern, Single Source Shortest Paths.

UNIT III

Dynamic Programming: General Method, Multistage Graphs, All Pair Shortest Paths, Traveling Salesperson Problem. Backtracking: General Method, N-Queen Problem, Sum of Subsets Problem, Graph Coloring, 0-1 Knapsack Problem.

UNIT IV

Branch & Bound: General Method, Least Cost Branch & Bound, 8-Queen Problem. Lower Bound Theory: Comparison Trees, Lower Bounds through Reductions, P & NP Problems, NP Hard & NP Complete Problems, Handling of NP Hard Problems. Parallel Algorithms: Basic Concept, Architectures, Effect of Parallelism.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Design and analyze programming problem statements.
- Understand the necessary mathematical abstraction to solve problems.
- Come up with analysis of efficiency and proofs of correctness.
- Comprehend and select algorithm design approaches in a problem specific manner.

Text Books:

1. Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press, Third Edition, 2009.

References:

1. Algorithms, by Dasgupta, Papadimitrou and Vazirani, McGraw-Hill Education, 2006.
2. Computer Algorithms, by Horowitz, Sahni, and Rajasekaran, Silicon Press, 2007.
3. Algorithm Design, by Kleinberg and Tardos, Pearson, 2005.
4. Algorithm Design, by Goodrich and Tamassia, Wiley, 2001.

Course Title:	Advanced Operating System	**Course Code:**	MCA611E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA554C	**Co-Requisite:**	MCA627E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand the basics of distributed systems.
- To understand the distributed communication using RPC
- To understand the various issue and challenges in during communication of distributed systems

COURSE CONTENT:

UNIT I

Introduction to Distributed Systems: Motivation & Goals, Advantages & Disadvantages of Distributed Systems over Centralized Systems, Hardware & Software Concepts. Design Issues: Transparency, Flexibility, Performance, Scalability, Heterogeneity, Reliability and Security. Concept of Client-server Model

UNIT II

Remote Procedure Calling (RPC): Motivation, Mechanism, Complexity, Design Issues & Classification. Synchronization in Distributed Systems: Physical Clock Synchronization (Centralized & Distributed Algorithms), Logical Clocks. Mutual Exclusion: Centralized & Distributed (Contention & Token) Algorithms. Election Algorithms: Bully Algorithm, Invitation Algorithm & Ring Algorithm.

UNIT III

Deadlock Detection: Centralized & Distributed Algorithms. Threads: Characteristics, Advantages & Disadvantages, Design Issues & Usage. System Models: Workstation Model, Processor Pool Model & Hybrid Model. Processor Allocation: Goals & Design Issues. File systems: Hierarchical file systems, Distributed File system (DFS): goals, Requirements, Components, Service Types. SUN-NFS: Design & Issues.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Provide hardware and software issues in modern distributed systems.
- Get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.
- Analyze the current popular distributed systems such as peer-to-peer (P2P) systems
- Know about Shared Memory Techniques.
- Design and Implement Distributed applications using Technologies like RPC, threads.
- Learn how to store data in Distributed File System.
- Understand How Distributed Shared Memory is managed.

Text Books:

1. A. S. Tanenbaum, “Distributed Operating systems”, Pearson Education, 2008.

References:

1. Couloris et. al., “Distributed Systems: Concepts and Design”, Pearson Education, 2005.
2. P. K. Sinha, “Distributed Operating Systems: Concepts & Design”, PHI Learning, 2007.

Course Title:	Distributed Databases	**Course Code:**	MCA612E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA502C	**Co-Requisite:**	MCA604C, MCA628E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies and showing the need for distributed database technology to tackle deficiencies of the centralized database systems.
- To expose active and emerging research issues in distributed database systems and application development.

COURSE CONTENT:

UNIT I

Features of Distributed versus Centralized Databases – Why Distributed Databases – Distributed Database Management Systems (DDBMSs)- Review of Databases – Review of Computer Networks-Levels of Distribution Transparency- Reference Architecture for Distributed Databases – Types of Data Fragmentation – Distribution Transparency for read-only Applications – Distribution transparency for Update Applications – Distributed Database Access Primitives – Integrity Constraints in Distributed Databases - A Framework for Distributed Database Design – The Design of Database Fragmentation – The Allocation of Fragments.

UNIT II

Foundations of Distributed Concurrency Control – Distributed Deadlocks – Concurrency Control Based on Timestamps – Optimistic Methods for Distributed Concurrency Control - Reliability – Basic Concepts Non-blocking Commitment Protocols – Reliability and Concurrency Control – Determining a Consistent View of the Network – Detection and Resolution of Inconsistency – Checkpoints and Cold Restart - Distributed Database Administration – Catalog Management in Distributed Databases – Authorization and Protection.

UNIT III

Distributed object database management systems – Fundamental object concepts and Models – Object – Abstract Data Types – Composition (Aggregation) – Class – Collection – Subtyping and Inheritance. – Object Distribution Design – Horizontal Class Partitioning – Vertical Class Partitioning – Path Partitioning – Class Partitioning Algorithms – Allocation – Replication – Alternative Client / Server Architectures – Cache Consistency – Object Identifier Management – Pointer Switching Object Migration – Distributed Object Storage – Object Query Processor Architectures – Query Processing Issues – Query Execution – Correctness Criteria – Transaction Models and Object Structures – Transactions Management in Object DBMSs – Transactions as Objects – Conclusion – Bibliographic Notes – Exercises.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design process.
- Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer.
- Describe distributed concurrency control based on the distinguished copy techniques and the voting methods
- Apply theory to practice by building and delivering a distributed database query engine, subject to remote Web service calls.

Text Books:

1. M.Tamer Ozsu, Patrick Valduriez, Distributed database systems, 2nd Edition, Prentice Hall of India, New Delhi.
2. Distributed Databases:Principles and Systems by Stefano Ceri, McGrawHill

References:

1. Distributed Database Systems by Chhanda Ray, Pearson
2. Distributed Database Management Systems: A Practical Approach by by Frank S. Haug and Saeed K. Rahimi, Wiley
3. Distributed Database Systems by David A. Bell, Jane B. Grimson, and Jane Grimson, Addison-Wesley

Course Title:	Bioinformatics	**Course Code:**	MCA613E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA502C,MCA551C,MCA529E,MCA579E	**Co-Requisite:**	MCA604C, MCA628E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To get introduced to the basic concepts of Bioinformatics and its significance in Biological data analysis.
- Describe the history, scope and importance of Bioinformatics and role of internet in Bioinformatics.
- Explain about the methods to characterize and manage the different types of Biological data.
- Classify different types of Biological Databases.
- Introduction to the basics of sequence alignment and analysis.
- Overview about biological macromolecular structures and structure prediction methods.

COURSE CONTENT:

UNIT I

Introduction to biology, Living cell: characteristics & functions, basics of molecular biology: DNA, RNA, gene, genetic code, processes of transcription, translation, splicing etc. ,basics of proteins, sequences, structure & functions of proteins, Introduction to bioinformatics: Definitions & concepts, & the role of bioinformatics, Types of biological data.

UNIT II

Introduction to databases, types of Biological Databases – flat file databases, relational databases, object-oriented databases, introduction to biological databases, Sequence databases (EMBL, GenBank, DDBJ, SWISS-PROT, PIR, TrEMBL), search engines: SRS,ENTREZ. BLAST, FASTA.

UNIT III

Basic overview of algorithms: Space & time complexity. Need & significance of sequence alignment. Global & local alignments, techniques of sequence alignments: Doplot, Needleman-wunsch algorithm for global alignment, Smith-waterman algorithm for local alignments, substitution matrices: PAM, Blosum etc. Gene prediction(ab initio & similarity based). Ontologies in bioinformatics: need for ontologies.

COURSE OUTCOMES:

At the end of the course student will be able to apply:

- knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics
- Existing software effectively to extract information from large databases and to use this information in computer modeling
- Problem-solving skills, including the ability to develop new algorithms and analysis methods
- An understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory, gene expression, and database queries

Text Books:

1. T. K. Attwood & D J Parry-Smith, “Introduction to bioinformatics”, Pearson Education

2. Jean-Michel Claveriw, CerdricNotredame, “Bioinformatics – A beginner’s Guide”, WILEY DreamTech India Pvt

References:

1. Harshawardhan P. Bal.” Bioinformatics Principles & Applications” O’Reily
2. T. K. Attwood & D J Parry-Smith, “Introduction to bioinformatics”, Pearson Education
3. Jean-Michel Claveriw, CerdricNotredame, “Bioinformatics – A beginner’s Guide”, WILEY DreamTech India Pvt
4. S.C. Rastogi, N. Mendiratta” Bioinformatics Methods &Applications”TMH
5. Krane ,”Bioinformatic”, Pearson Education
6. B. Bergeron, “Bioinformatics Computing”, Pearson
7. D. E. Krene& M.L. Payma, “Fundamental concepts of Bioinformatics”, Pearson

Course Title:	System Software Design	**Course Code:**	MCA614E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA551C, MCA563E	**Co-Requisite:**	MCA611E, MCA627E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVE:

- The objective of this course is to teach procedures for the design of software systems and to provide a basis for judgement in the design of system software. The focus is to discuss the design and implementation of the major system software components with the consideration of underlining the pertinent design issues. The design issues are likely to be discussed in the context of modern computer systems that include the different languages and advanced operating systems.

COURSE CONTENT:

UNIT I

Overview, Introduction to Machine Structure, Evolution of the System Programming Components: Assemblers, Loaders, Macros, Linkers, Compilers.

Evolution of Operating System & User View: Functions, Batch Control Language & Facilities.

Machine Structure 360-370: Memory, Registers, Data, Instructions & Special Features. Machine Language & Assembly Language.

UNIT II

Assemblers: General Design Procedure: Problem Statement, Data Structures, Format of Databases, Algorithms, Modularity Lookup. Table Processing: Review of Searching & Sorting Techniques. Macros: Macro Instructions, Features, Conditional Macro Instructions, Macro Calls within Macros, Single & Two Pass Algorithm, Implementation of Macro Calls within Macros. Implementation within an Assembler

UNIT III

Loaders: Schemes: Compile & Go Loaders, General Loader Scheme, Absolute Loaders, Subroutine Linkages, Relocating Loaders. Other Loader Schemes & Binders: Linking Loaders, Overlays & Dynamic Binders. Design of an Absolute Loader, Design of a Direct Linking Loader. Programming Languages & Formal Systems: Significance of High Level Languages, Features. Data Types & Data Structures, Storage Allocation & Scope of Names, Accessing Flexibility, Functional Modularity, Use of Formal Systems, Formal Specification, Formal Grammars, Hierarchy of Languages, Backus Normal Form – BNF, Canonic Systems & Formal Systems.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Basic Understanding of Machine Structure and its Major System Software Components
- Comprehend Assembler Design Issues and Procedures
- Understand Macros and Macro Challenges
- Understand various Loader Schemes with the consideration of underlining the pertinent design issues
- Understand Compilers in the context of different languages and advanced operating systems.

Text Books:

1. John J. Donovan, “Systems Programming”, Tata McGraw-Hill

References:

1. Barron.D.W. "Assemblers & Loaders", Mc Donald & Javes
2. Ullman.J.D."Fundamentals of Programming System" Addison & Wesley.
3. M.Dhamdhere."System Programming & Operating Systems"

Course Title:	Natural Language Processing	**Course Code:**	MCA615E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA552C, MCA579E	**Co-Requisite:**	MCA602C,MCA614E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand natural language processing and to learn how to apply basic algorithms in this field.
- To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora.
- To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

COURSE CONTENT:

UNIT I

The study of Language, challenges of NLP, The Different Levels of Language Analysis, Representations & Understanding, Linguistic background, Grammars & Parsing, Features & Augmented Grammars, Grammars for Natural Language, Towards Efficient Parsing, Ambiguity Resolution. Applications in Information Retrieval, design features of information retrieval system, Language Modelling: Grammar based model & statistical model.

UNIT II

Word Level & Syntactic Analysis: Spelling error detection & correction, parts of speech tagging Morphological parsing, Probabilistic parsing. Semantic Interpretation: Semantics & Logical Form, Linking Syntax & Semantics, Strategies for Semantic Interpretation. Pragmatics: Reference Resolution, Syntactic & Semantic Coherence, Text Coherence, An Inference based resolution algorithm.

UNIT III

Natural language Generation: Introduction to language generation, architecture for generation, surface realization, systemic grammar, functional unification grammar, discourse planning Machine Translation: problems in machine translation, Language Similarities & Differences, transfer metaphor, syntactic transformations, lexical transfer, idea of Interlingua, direct translation, using statistical Techniques.

COURSE OUTCOMES:

At the end of the course student will be able to:

- The students will get acquainted with natural language processing and learn how to apply basic algorithms in this field.
- They will understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language processing.
- They will also grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

Text Books:

1. James Allen, "Natural Language Understanding", Pearson Publication, ISBN: 978-81-317-08958 2nd Edition

References:

1. D. Jurafsky, J. H. Martin, “Speech & Language Processing”, Pearson Education, 2002.
2. Speech & Language Processing, by Jurafsky, D. & Martin,
3. J.H.Tanveer Siddiqui, US Tiwary, Natural Language Processing & Information Retrieval

Course Title:	Digital Image Processing	**Course Code:**	MCA626E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA552C, MCA579E	**Co-Requisite:**	MCA602C, MCA614E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To study image fundamental and mathematical transforms necessary for image processing.
- To study fundamental technologies for digital image, compression, analysis, and processing
- To Gain understanding of algorithm, analytical tools, and practical implementations of various digital image applications

COURSE CONTENT:

UNIT I

Introduction & Digital Image Fundamentals: The origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamentals Steps in Image Processing, Elements of Digital Image Processing Systems, Image Representation, Image Sampling & Quantization, Some basic relationships like Neighbors, Connectivity, Distance Measures between pixels, Linear & Non Linear Operations. Image Enhancement in the Spatial Domain: Some basic Gray Level transformations, Histogram Processing, Enhancement Using Arithmetic & Logic operations, Basics of Spatial Filters, Smoothening & Sharpening Spatial Filters, Combining Spatial Enhancement Methods.

UNIT II

Image Enhancement in the Frequency Domain: Introduction to Fourier Transform & the frequency Domain, Smoothing & Sharpening Frequency Domain Filters, Homomorphism Filtering. Image Restoration: A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Dedradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.

UNIT III

Image Compression: Coding, Interpixel & Psychovisual Redundancy, Image Compression models, Elements of Information Theory, Error free comparison, Lossy compression, Image compression standards. Image Segmentation: Detection of Discontinuities, Edge linking & boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand Basics of Image formation and transformation using sampling and quantization.
- Perform and apply compression and coding techniques used for image data.

Text Books:

1. Rafael C. Gonzalez &amp; Richard E. Woods, “Digital Image Processing”, Pearson Education

References:

1. A.K. Jain, “Fundamental of Digital Image Processing”, PHI
2. Bernd Jahne, “Digital Image Processing”, Springer
3. William K Pratt, “Digital Image Processing: Piks Inside”, John Wiley & Sons

List of Practicals:

LAB JOURNAL_ MCA626E	
S. No.	**TITLE**
1.	**Image Representation & Sampling** Convert an image into its digital representation (e.g., grayscale or RGB). Demonstrate the effect of different sampling rates on image quality.
2.	**Image Quantization & Basic Operations** Quantize the image using different bit-depths. Perform basic operations like addition, subtraction, and multiplication on images.
3.	**Neighbors, Connectivity, & Distance Measures** Implement algorithms to find neighboring pixels. Calculate distance measures like Euclidean distance between pixels.
4.	**Linear & Non-Linear Operations** Implement linear operations like contrast stretching and gamma correction. Implement non-linear operations like thresholding and histogram equalization.
5.	**Spatial Filters** Implement basic spatial filters like mean and median filters. Demonstrate the effects of different filter sizes on image quality.
6.	**Combination of Spatial Enhancement Methods** Combine spatial enhancement methods (e.g., filtering followed by histogram equalization) and compare results.
7.	**Fourier Transform & Frequency Domain Filters** Implement the 2D Fourier transform. Apply frequency domain filters like low-pass and high-pass filters.
8.	**Image Restoration Techniques** Implement inverse filtering for deconvolution. Implement Wiener filtering for noise reduction.
9.	**Periodic Noise Reduction** Implement frequency domain filtering to reduce periodic noise. Demonstrate the effect of different filter parameters on noise reduction.
10.	**Geometric Transformations** Implement basic geometric transformations like rotation and scaling. Apply geometric transformations to images and analyze the results.
11.	**Image Coding & Compression** Implement basic coding and compression techniques like Huffman coding. Analyze the compression ratio and image quality.
12.	**Psychovisual Redundancy** Analyze the psychovisual redundancy of different image regions. Demonstrate the effect of psychovisual redundancy on compression efficiency.
13.	**Image Segmentation Techniques** Implement edge detection algorithms like Sobel and Canny. Implement region-oriented segmentation using clustering algorithms.
14.	**Motion-Based Segmentation** Implement motion-based segmentation using background subtraction techniques. Analyze the effect of different motion models on segmentation accuracy.

Course Title:	Linux System Programming	**Course Code:**	MCA627E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA554C	**Co-Requisite:**	MCA611E, MCA614E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVE:

- The objective of course Linux System Programming is to help users in development process and thus maximize the programming time and applications use of Linux system and encourages and enable the users to develop their own real-world applications.

COURSE CONTENT:

UNIT I

Introduction Linux, History of Linux, Linux Architecture & features, Linux and Unix. Introduction to Linux file System. File operations, Mounting and Unmounting of file systems, Different files types (Ext2 & Ext3). Files & Directory Structure. Overview of Linux process management, process states. Linux Editors and Compilers, Overview of YACC and LEX.

UNIT II

Introduction to Shell programming: Command line programming, Shell types (Bourne Shell, Korn Shell, C shell), Shell Script and structure, Running a Shell Script, Introduction to Linux programming, declaring Variables, input and output operations, Using break, continue, exit, return & Special Parameters ($* and $@) Operators, Conditional Constructs IF-Then-FI,IF-THEN-ELIF-FI, Looping Constructs: for-in, while, untill , Built-in Mathematical Functions,

UNIT III

Filters and Shell Command: Pr, Head, tail, cut, paste, sort, uniq, tr, join etc. grep, egrep, fgrep, sed, awk etc. File permission commands, File and directory command. Pipeline & Input/output redirection and commands, Introduction to system calls, creat, open, close, read etc. background and foreground process managing commands. Linux GUI Development: Introduction to Qt as GUI development tool, Managing Controls and forms in Qt.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Use basic fundamental utilities which are required again and again on daily basis to work on a modern operating system.
- Write useful shell scripts which greatly and effectively enhance the usefulness of computers, from the point of view of programmers and application developers.
- Understand basics of various OS related concepts, from programmer's point of view, like files, directories, kernel, inodes, APIs, system calls, processes, signals, etc.
- Develop applications where several processes need to communicate with each other to complete a task.
- Use different IPC ways in their programs like Message Queues, Semaphores, and Shared Memories.
- Write programs which employs advanced concepts like multithreading.
- Write useful programs for networking purposes

Text Books:

1. A practical guide to Linux commands, Editors and Shell Programming by Mark G.Sobell.
2. System Programming n Linux/Unix by K.C Wang publisher:Springer International
3. Understanding the Linux Kernal Publisher O'Relly Media.

References:

1. Linux System Programming Talking Directly to the Kernel & C Library By Robert LovePublisher: O'Reilly Media Final Release Date: September 2007 Pages: 392
2. Advanced Programming in Unix Environment Second Edition W.Richard Steven Stephen A Rago Publisher Addison Wesley

List of Practicals:

LAB JOURNAL_ MCA627E	
S. No.	**TITLE**
1.	Input the distance in KMS and WAP to convert and print it in meters, feet
2.	Calculate the area and perimeter of a Rectangles ad area and circumference of a Circle.
3.	Write a shell script to enter a name and Check whether it is file or directory. If it is a file display its permissions.
4.	Write a shell script to calculate the sum of digits of a three digit number.
5.	Write a program whether the seller has profit or loss in selling on item. Also show how much loss or profit he has incurred.
6.	Write a shell Script to determine whether the number is even or odd.
7.	Write a shell script to check whether an input number is within range from 1 to 50 or not.
8.	Write a program to receive two filenames as arguments and check whether their contents are same or not if same delete the second file.
9.	Write a shell script to enter a word and determine whether it begins with a same case, Vowel or Capital vowel or ends with a digit or it is a three lettered word.
10.	Write a Shell Script to perform real number calculation in Shell Script and Store result to third variable.
11.	Write a Shell Script to find factorial of Number.
12.	Write a Shell Script to generate all prime Numbers from 1 to 50.
13.	Write a shell script to cont the no. of lines,words and Characters supplied at standard I/O.
14.	Write a Shell Script to find out biggest number from their numbers and numbers are supplied at command line.
15.	Write a Shell Script to generate Fibonacci series upto n terms.
16.	Write a Shell Script to print all combinations of 1 2 3.
17.	Write a Shell Script which will receive any number of filenames as arguments. The shell script should check whether every argument is file or a directory. Also count the no. of lines.
18.	Write a Shell Script to print the given number in Reverse Order.
19.	Write a Shell Script to see the current date, time, Username and current
20.	Write a Shell Script to copy contents of source file to target file and message should be displayed if source is unavailable.
21.	Write a Shell Script to generate Armstrong series from 1-1000.
22.	Write a Shell Script to check Whether a given number is perfect or not.
23.	Write a Shell Script to check whether a year is leap or not.
24.	Write a Shell Script to print multiplication table for any number using for loop.
25.	Write a Shell Script to print perfect number form 1-100.

26.	Write a Shell Script to demonstrate the use of following commands: a. PS to view the process attributes, b. Kill to terminate process c. Nice to reduce the priority of a job d. Grep,egrep and fgrep.
27.	Write a Shell Script to explain the Ownership and file permissions.
28.	Demonstrate the use of chmod command to change file permissions.
29.	Write a Shell Script to demonstrate the use of file handling Unix commands

Course Title:	Data Warehousing	**Course Code:**	MCA628E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA502C, MCA511C	**Co-Requisite:**	MCA612E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To provide students with an understanding of the fundamentals of data warehouse, its basic components and the role of data warehouse in decision support.
- To provide knowledge of the important steps and techniques to be considered during data warehouse development, future trends and usage of data warehouse.
- To introduce the concept of dimensional modelling technique for designing a data warehouse.
- To provide knowledge of data integration and the extraction, transformation and load (ETL) processes.
- To introduce students to the methods and tools for accessing and analyzing the warehouse data.

COURSE CONTENT:

UNIT I

Overview of decision support systems: Organizational need for strategic information, Failures of past decision-support systems, operational versus decision-support systems, data warehousing – the only viable solution, data warehouse defined. Data warehouse – The building Blocks: Defining Features, data warehouses & data marts, overview of the components, metadata in the data warehouse

Defining the business requirements: Dimensional analysis, information packages – a new concept, requirements gathering methods, requirements definition: scope & content

UNIT II

Principles of dimensional modeling: Objectives, From Requirements to data design, the STAR schema, STAR Schema Keys, Advantages of the STAR Schema

Dimensional Modeling: Updates to the Dimension tables, miscellaneous dimensions, the snowflake schema, aggregate fact tables.

UNIT III

Data Extraction, Transformation and Loading: Objectives, ETL Overview, ETL Requirements and Steps, Data Extraction- Source Identification, Data Extraction Techniques, Evaluation of Techniques, Data Transformation-Major Transformation Types, Data Integration and

Consolidation, Transformation for Dimension Attributes, Transformation Implementation, Data Loading, ETL Tool Options, Other Integration Approaches.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Understand the concept related to data warehouse architecture
- Compare Online Analytical Processing (OLAP) and Online Analytical Transaction Processing(OLTP) tools
- Design a data mart or data warehouse for any organization
- Asses raw input data and preprocess it to provide suitable input for range of data mining algorithms
- Extract association rules and classification model
- Identify the similar objects using clustering techniques

Text Books:

1. Paul Raj Poonia, “Fundamentals of Data Warehousing”, John Wiley & Sons

References:

1. Sam Anahony, “Data Warehousing in the real world: A practical guide for building decision support systems”, John Wiley
2. Alex Berson,Stephen J. Smith, “Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill W. H. 5.
3. Inmon, “Building the operational data store”, John Wiley
4. Kamber & Han, “Data Mining Concepts & Techniques”, Hartcourt India P. Ltd.

List of Practicals:

LAB JOURNAL_ MCA628E	
S. No.	**TITLE**
1.	Based on a given case study: a. Define business requirements b. Perform Dimension Analysis c. Design information packages d. Do requirement gathering
2.	Based on a given case study: a. Design of fact & dimension tables b. Generating graphs for star schema c. Generation of snowflake schema
3.	Identify source tables and populate sample data.
4.	Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking-Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc.).
5.	Write ETL scripts and implement using data warehouse tools
6.	Perform Various OLAP operations such slice, dice, roll up, drill up and pivot.
7.	Explore visualization features of the tool for analysis like identifying trends etc.
8.	Explore WEKA Data Mining/Machine Learning Toolkit

Course Title:	Open Source Technologies	**Course Code:**	MCA629E
Semester:	3rd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA55,MCA526E,MCA529E,MCA579E,	**Co-Requisite:**	MCA601C,MCA603,MCA

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To learn about Linux Administration.
- To learn PHP and its database connectivity.
- To learn how to create Websites with PHP MY-SQL.
- To learn Content Management System. And Introduction to No-Sql.

COURSE CONTENT:

UNIT I

Introduction to Linux – Architecture, Installation, Configuration, Package Management, Basic Linux commands - file system commands, vim. Apache Http Server – installation, basic configuration, setting up virtual hosts, installation of modules, etc.

UNIT II

Introduction to PHP - variables, data types, basic constructs, loops, functions, classes. Installation of third party php modules. Introduction to MySQL database. PHP-MySQL connectivity. Introduction to HTTP – HTTP methods, headers, cookies, etc. Creating a basic website with PHP & MySQL.

UNIT III

Content management system based on php & mysql e.g. Wordpress & Drupal – installation, configuration. Introduction to NoSQL – installation & configuration of Mongodb. Basic select, insert, update & delete operations, aggregate operations. Backup & restoration.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Explore different open source technology like Linux, PHP & MySQL with different packages.
- Execute Linux commands for programming.
- Execute programs of PHP with MySQL connection

Text Books:

1. Core Java Volume 1-Fundamentals by Cay S.horstman 10th edition Publisher: Prentice Hall

References:

1. Java the complete reference by Herbert Schield 10th edition Publisher:Tata Mc Graw Hill
2. Java How to program Dietel and Dietel

List of Practicals:

LAB JOURNAL_ MCA629E	
S. No.	**TITLE**
1.	**Linux Architecture & Installation** Install a Linux distribution (e.g., Ubuntu, CentOS). Explore the Linux file system.
2.	**Linux Package Management** Use package managers like apt or yum to install and update software packages.
3.	**Basic Linux Commands & Vim** Practice basic file system commands like ls, cd, mkdir, etc. Learn basic Vim commands for text editing.
4.	**Apache Http Server** Install Apache HTTP Server. Configure basic settings like port number and document root.
5.	**Apache Virtual Hosts** Set up virtual hosts for multiple websites on a single server.
6.	**Introduction to PHP** Learn about PHP variables, data types, and basic constructs. Practice writing loops, functions, and classes in PHP.
7.	**Installation of PHP Modules** Install third-party PHP modules (e.g., PHP-GD for image processing).
8.	**Introduction to MySQL Database** Install MySQL Database Server. Perform basic operations like creating databases and tables.
9.	**PHP-MySQL Connectivity** Write PHP scripts to connect to and interact with a MySQL database.
10.	**Introduction to HTTP** Learn about HTTP methods, headers, and cookies. Practice sending HTTP requests and handling responses.
11.	**Creating a Basic Website with PHP & MySQL** Develop a simple website using PHP for server-side scripting and MySQL for database storage.
12.	**Content Management Systems (CMS)** Install and configure a CMS like WordPress or Drupal. Explore the admin interface and create basic content.
13.	**Introduction to NoSQL** Install and configure MongoDB. Practice basic operations like inserting, updating, and deleting documents.
14.	**NoSQL Aggregation Operations** Learn about MongoDB's aggregation framework. Practice using aggregation operations like $match, $group, and $sort.
15.	**Backup & Restoration** Perform backups of your MySQL and MongoDB databases. Practice restoring databases from backups.

Semester - IV

Semester-IV (24 Credit Semester and 32 hours per week)

Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MCA651C	Theory of Formal Languages	Core	4	3	1	0	04	FC
MCA652C	Big Data Analytics	Core	4	3	1	0	04	AEC
MCA653C	Project work	Core	10	1	2	7	17	EOC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MCA661E	Wireless Communications	DCE	3	2	1	0	03	SEC
MCA662E	Cloud and Grid Computing	DCE	3	2	1	0	03	SEC
MCA663E	Information Security and Networks	DCE	3	2	1	0	03	SEC
MCA664E	Compiler Design	DCE	3	2	1	0	03	AEC
Pool B: 03 Discipline Centric Elective Credit Units								
MCA676E	Block Chain Technologies	DCE	3	2	0	2	04	EC
MCA677E	Quantum Computing	DCE	3	2	0	2	04	AEC
MCA678E	Advanced Java	DCE	3	2	0	2	04	EC
MCA679E	Deep Learning	DCE	3	2	0	2	04	SEC
Total			**24**				**32**	

Course Title:	Theory of Formal Languages	**Course Code:**	MCA651C
Semester:	4th	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA615E, MCA503C, MCA512E	**Co-Requisite:**	MCA664E

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To be able to construct finite state machines and the equivalent regular expressions.
- To understand the class of languages described by finite state machines and regular expressions.
- To understand and construct pushdown automata and the equivalent context free grammars.
- To understand the concept of construct Linear bounded automata and the equivalent context Sensitive grammars.
- To understand the concept of Turing Machine and Universal Turing machine

COURSE CONTENT:

UNIT I

Introduction to Set Theory, Introduction to Automata, Alphabets, String, Languages, Strings, Kleen Star, Kleen Plus, Introduction to Finite automata, Deterministic finite automata, DFA notations (Transition Table, Transition Table) Languages Accepted by DFA, Designing DFA by pattern recognition, Application of DFA, Non-deterministic Finite Automata, Language accepted by NDFA, Conversion of DFA to NDFA, Overview of e-NDFA. DFA vs NDFA vs e-NDFA.

UNIT II

Finite Automata & Regular Expressions: Definition, Basic Regular Expressions, Obtaining regular expression: Using Basics Expressions & State Elimination Method, Application of regular Expressions, Regular Languages: Definition, Properties, and Pumping Lemma for Regular Languages, Decidable and Closure properties of Regular languages. Limitation of Finite Automata, Equivalence & minimization of DFA.

UNIT III

Introduction to Grammar, Chomsky Hierarchy, Generation of Grammar from Finite Automata & Regular Expressions, Derivation Trees, Left most and Right Most Derivation Trees, Ambiguous Grammar, Context Free Grammar and its applications, properties of context free languages. Introduction to Push down Automata, Transition & Graphical Representation.

UNIT IV

Introduction to Linear bounded Automata & Turing Machine. Turing machine model, transition table Instantaneous definition, Acceptance and Construction of Turing machine. Standard Turing machine, Transducers, Lexical and Syntax Analyze: Parsing, Top down Parsing (Predictive parsers, LL (K), recursive Descent) and Bottom up Parsing (LR (K)).

COURSE OUTCOMES:

At the end of the course student will be able to:

- Describe the mathematical model of machines.
- Understand the concept of formal language and corresponding automaton.
- Introduces the concept of ambiguity, derivations and parse tree in grammar.
- Apply the acquired knowledge of formal languages to the engineering areas like Compiler Design
- Apply the acquired knowledge of finite automata theory and to design discrete problems to solve by computers.

Text Books:

1. Finite Automata and Formal Languages"- A Simple Approach, A.M Padma Read Pearson Education

References:

1. Hopcroft, J., & Ullman, J., "Introduction to Automata Theory, Languages & Computation", Pearson Education
2. Hopcroft J, R. Motwani, & J. Ullman, "Introduction to Automata Theory, Languages & Computation", Pearson Education
3. P. Linz, "Introduction to Formal Languages & Automata", Jones &Barlett PWS Publishing Company
4. Donald Knuth, "The Art of Computer Programming", Prentice Hall

Course Title:	Big Data Analytics	**Course Code:**	MCA652C
Semester:	4th	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA502C, MCA552C	**Co-Requisite:**	MCA662E

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To learn big data challenges in different domains including social media
- To learn MAP-REDUCE programming model for better scalability and performance
- To analyze the capability of No-SQL systems
- To apply machine learning algorithms for big data analytics

COURSE CONTENT:

UNIT I

Introduction to Big Data: Big Data - Definition, overview of Big Data, Big Data Characteristics Volume, Velocity, Variety and other Vs, Issues and challenges of Big Data, Stages of analytical evolution, Big Data Architecture, Space of Big Data: Transactions, Interactions, Observations; Big data Technological approaches and Potential use cases for Big Data. Storage: A General Overview of High-Performance Architecture; HDFS, MapReduce, Map Reduce Programming Model, Overview of HBase, Hive and PIG. Spark Framework – Spark Data frames, Importing and saving data.

UNIT II

Data collection and management: Introduction, Sources of data, Data collection and APIs, data quality and issues with date collection systems with examples from business; Cleaning and treatment of missing data, principles of data visualization, and different methods of presenting data in business analytics. Introduction to Statistics - Measures of central tendencies and distributions, Analysis of variance, Correlation analysis. Distribution properties and arithmetic. Basic analysis techniques - Statistical hypothesis generation and testing.

UNIT III

Machine Learning: Introduction, Supervised Learning, Unsupervised Learning. Classification: Decision Trees - Overview of a Decision Tree, Evaluating a Decision Tree, Confusion matrix. NoSQL Databases Review of traditional Databases, Need for NoSQL Databases, Columnar Databases, Failover and reliability principles, working mechanisms of NoSQL Databases HBase, Cassandra, Couch DB, Mongo DB.

UNIT IV

Data Science - Introduction, Terminology, data science process, data science toolkit. Applications of Data Science, Data Formats – CSV/JSON/AVRO/Parquet. Data Visualization - Technologies for visualization, Tools and libraries for Big Data visualization (e.g., Matplotlib/Seaborn/Tableau/Power-BI). Effective communication of Big Data insights through visualizations.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Preparing for data summarization, query, and analysis.
- Applying data modelling techniques to large data sets
- Creating applications for Big Data analytics
- Building a complete business data analytic solution

Text Books:

1. Big Data – A Primer, H. Mohanty, P. Bhuyan, D. Chenthati (Eds.), Springer , Studies in Big Data, vol. 11, 2015.
2. Big Data Analytics with Rand Hadoop, Vignesh Prajapati, PACKT Publishing

References:

1. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Cambridge Universities Press, 2012.
2. Big Data at Work: Dispelling the Myths, Uncovering the Opportunities Book by Thomas H. Davenpor

Course Title:	Project Work	**Course Code:**	MCA653C
Semester:	4th	**Paper Type:**	Core
Credits:	10	**Max Marks:**	250
Pre Requisite:	All courses upto semester 3rd	**Co-Requisite:**	MCA-4T2-C, MCA-4E8-DCE

Marks Distribution: (Mid Term: 100, End Term: 150, Viva: 0, Assignment / Presentations:0)

PROJECT WORK

Course Title:	Wireless Communications	**Course Code:**	MCA661E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA541A, MCA603C,MCA511E	**Co-Requisite:**	MCA663E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The objective of this course is to give an introduction to the fundamentals of the wireless communications systems, the wireless network architectures, protocols, and applications.
- To provide an overview of Wireless Communication networks area and its applications in communication engineering.
- To enable the student to synthesis and analyze wireless and mobile cellular communication systems over a stochastic fading channel
- To provide the student with an understanding of advanced multiple access techniques.
- To provide the student with an understanding of diversity reception techniques
- To give the student an understanding digital cellular systems (cdma2000, W-CDMA, LTE and candidate 5G waveforms).

COURSE CONTENT:

UNIT I

Wireless Communication Systems: Basic Concepts, Wireless Networks, Wireless vs Wired networks, Classification of Wireless Networks. Wireless Transmission Concepts: Antennas, Types of antennas, Signal propagation, Multipath propagation. Types of Wireless Telephones (Cordless, Fixed Wireless (WLL), Wireless With Limited Mobility (WLL-M), Fully Mobile Wireless Phones). Introduction to various Generations of Mobile Phone Technologies. Wireline/Wireless Portion of Mobile Communication Network.

UNIT II

Electromagnetic spectrum, its use & allocation to well-known bands. Multiple Access Techniques: FDMA, TDMA & CDMA. Basic concepts of Spread Spectrum(SS) technique; Direct Sequence SS versus Frequency Hopping SS. Concept of cells, sectorization, coverage area, frequency reuse, cellular networks, microcells & handoffs. Simplified Implementation of IS-95 CDMA Using Chip Sequences, Concept of (Walsh) Code, Concept of CDMA (PCS and Cellular) Channel, Forward & Reverse CDMA Channel for a Cell/Sector, Comparison of Cellular/PCS CDMA Networks, Frequencies & Cell-Sizes.

UNIT III

Cellular Mobile Systems: Introduction & Overview of GSM/CDMA. Channel: Types, Forward & Reverse CDMA channel for a cell/sector (Pilot, Sync, Paging, Forward Traffic Channels, Access, Reverse TC). Concept of Voice Coding. Components of Mobile Network Architecture: MS, BTS, BSC, MSC, & their basic Functions. Types of Handoffs in GSM. Use of HLR & VLR in Mobile

Networks. Concept of Mobile IP & WAP. Wireless LAN Technology: Overview of Wifi & the IEEE 802.11. Concept of Bluetooth & IEEE 802.15.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases.
- Demonstrate knowledge of protocols used in wireless networks and learn simulating wireless networks.

Text Books:

1. T. Rappaport, “Wireless Communications, Principles andamp; Practice”, Pearson andy Dornan, “The Essential Guide to Wireless Communications”, Pearson
2. Jochen Schiller, “Mobile Communications”, Pearson andrew Tanenbaum, “Computer Networks”, PHI

References:

1. William Stallings, “Wireless Communications & Networks”, Pearson
2. K.Pahlavan, P.Krishnamurthy, “Principles of Wireless Networks”, Pearson Education.

Course Title:	Cloud & Grid Computing	**Course Code:**	MCA662E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA603C, MCA612E,MCA614E, MCA629E	**Co-Requisite:**	MCA663E,MCA676E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVE:

- The objective of studying Cloud and Grid Computing is that it enables new social services by connecting users via social networks that are constructed using multiple cloud service and it will run the software industry upside down. Thus educates potential users about the benefits of cloud computing and the best way to make full use of them.

COURSE CONTENT:

UNIT I

Cloud Computing: Introduction to cloud & Cluster computing, cloud computing vs cluster computing, Evolution of cloud computing, principles of cloud computing, cloud Computing architecture, Cloud computing applications. Cloud service models (IaaS, PaaS, SaaS), could Deployment models (Public, Private, hybrid& community models). Challenge and Security Issues

UNIT II

Grid Computing: Introduction to Grid Computing, Characteristics, grid computing Architecture, grid complexity levels and topologies, grid components and grid layers, applications of Grid Computing. Grid security issues: Authorization and Authentication methods, Grid computing vs Cloud Computing Cluster Computing.

UNIT III

Introduction to OGSA, Services, Schema and architecture, Overview of OGSI, Virtualization: Virtual machines and visualization of clusters and data centres, levels of virtualization, virtualization structures and tools and mechanism, virtualization of CPU, Memory & I/O devices

COURSE OUTCOMES:

At the end of the course student will be able to:

- Elaborate the basic concepts of cloud computing and defining the basic terms
- Understand various cloud implementations and migration techniques
- To define the various industrial applications of cloud virtualization.
- learn security challenges and preventive measures in cloud computing

Text Books:

1. Cloud Computing by pankaj Sharma published by S.K.Kataria and Sons.
2. Cloud Computig Bible:Published by Wiley India Pvt. Ltd.
3. Grid and Cloud Computing and Applications: The 2014 World Comp InternationalConference.Publisher C.S.R.E.A
4. Grid, Cloud and Cluster Computing and Applications by Hamid R.Arabnia, Ashu M.G.Solo

References:

1. Mastering Cloud Computing, Paperback-1 Feb 2013 by Buyya, Vecchiola&Selvi.
2. Cloud Computing: Concepts, Technology & Architecture, 1e Paperback-2014 by Erl.
3. Grid & Cloud Computing, a business perspective on Technology & Applications, Springer by Stanoevska-Slabeva, Katarina, Wozniak, Thomas, Ristol, Santi (Eds.)

Course Title:	Information Security & Networks	**Course Code:**	MCA663E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA513E, MCA603C	**Co-Requisite:**	MCA661E,MCA662E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.
- To Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.
- Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

COURSE CONTENT:

UNIT I

Introduction: Security trends, the OSI Security Architecture, Security attacks, Security mechanisms, Security services, Model for Network Security. Encryption: Symmetric & Asymmetric Encryption, Symmetric Cipher model, Substitution techniques (Mono-alphabetic & Poly-alphabetic) & Transposition Technique, Rotor Machines, Block and Stream ciphers, Modern Round Ciphers: DES, AES, Introduction to Steganography. Encryption vs Steganography.

UNIT II

Introduction to Number Theory: Prime Numbers, Fermat's & Euler's theorem & discrete logarithms. Public Key Cryptography & RSA. Key management: Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic & Elliptic Curve Cryptography. Message Authentication & Hash functions. Digital Signatures & Authentication protocols. Network Security Applications: IP Security

UNIT III

Mechanics of Routing Protocols: Routing , Static and Dynamic Routing, Distant vector routing & link state routing, Internet working with Dissimilar Protocols, Protocol design Consideration: Simplicity, flexibility, Optimality, Overhead & Scaling, Operation above Capacity, forward compatibility. Migration: Routing Algorithms & addressing parameters, making multi-protocol operation possible.Network Security: Security Features in Wireless Adhoc & Wireless Sensor Networks.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Evaluate the security treats in modern computer era
- Define and identify firewall and network filtering
- List and recognize various VPN
- Identify different technique of sandboxing
- Distinguish various ethical hacking and testing procedures

Text Books:

1. William Stalling ," Cryptography & Network Security", Pearson Education

References:

1. Radia Perlman, "Interconnections: Bridges, Routers switches & Internet-working protocols Pearson education
2. Mark Sportack, "IP Routing Fundamentals", Pearson Education
3. Gerard J. Holzmann, "Design & Validation Computer Protocols", Prentice Hall

Course Title:	Compiler Design	**Course Code:**	MCA664E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA551C, MCA615E	**Co-Requisite:**	MCA651C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand the various phases in the design of a compiler.
- To understand the design of top-down and bottom-up parsers.
- To understand syntax directed translation schemes.
- To learn to develop algorithms to generate code for a target machine.

COURSE CONTENT:

UNIT I

Introduction to compilers: Structure of a compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering –Specification of Tokens – Recognition of Tokens – Lex – Finite Automata – Regular Expressions to Automata – Minimizing DFA.

UNIT II

Syntax Analysis: Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar – Top Down Parsing – General Strategies Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR(0)Item Construction of SLR Parsing Table -Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer

Intermediate Code Generation: Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

UNIT III

Run-Time Environment And Code Generation: Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management-– Issues in Code Generation – Design of a simple Code Generator.

Code Optimization: Principal Sources of Optimization – Peep-hole optimization – DAG- Optimization of Basic Blocks Global Data Flow Analysis – Efficient Data Flow Algorithm.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Get an understanding of how the different phases of compiler work.
- Study of lexical analysis, and various parsing techiques.
- Use compiler tools like lex and yacc.
- Implement a simple compiler for a hypothetical machine using syntax directed translation

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson.
2. Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning.

References:

1. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
2. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH

Course Title:	Block Chain Technologies	**Course Code:**	MCA676E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA603C, MCA604CMCA612E, MCA629E	**Co-Requisite:**	MCA652C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To understand the function of Block chain as a method of securing distributed ledgers
- To familiarize the functional/operational aspects of cryptocurrency ecosystem.
- To familiarize about wallets and learn their utilization of wallet during transaction
- To understand that how to write and apply the Smart Contracts
- To understand the concept of Hyperledger

COURSE CONTENT:

UNIT I

Introduction to Cryptography, Introduction to graph, ring and field, prime and relative prime numbers, modular arithmetic, Fermat's and Euler's theorem, Euclid's Algorithm, RSA algorithm, Diffie-Hellman key exchange algorithm, ElGamal Encryption, Elliptic curve cryptography, SHA 256, Digital Signature, Zero Knowledge Proof (ZKP)

UNIT II

Introduction from barter system to Cryptocurrency, fundamental of Blockchain, Block structure, Genesis Block, Orphaned Blocks, Stale Block, Uncle Block, Distributed Ledger Technology (DLT), peer-to- peer network, Merkle Tree, Lifecycle of Blockchain, Evolutions of Blockchain, Fork, double spending money, Transactions and UTXO's, Types of Blockchain. Need of Blockchain, Benefits of Blockchain

UNIT III

Cryptocurrencies: BitCoin (BTC), Ethereum (ETH), Ripple (XRP), LiteCoin (LTC), Bitcoin Cash (BCH),Mining pools, Mining, Difficulty Level, Current Target, Nonce, how miners picks transactions,How do mempools work, 51% attack.

Consensus Algorithms: Proof of Work (PoW), Asynchronous Byzantine Agreement, Proof of Stake (PoS), Hybrid models (PoW + PoS).

COURSE OUTCOMES:

At the end of the course student will be able to:

- Implement the blockchain
- Implement the smart contracts on Ethereum platform.
- Implement the use cases on Hyperledger
-

Text Books

1. Mastering Blockchain, Imran Bashir, Packt Publishing
2. Bitcoin and Cryptocurrency Technologies, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Princeton University Press. https://bitcoinbook.cs.princeton.edu/

References:

1. Grokking Bitcoin, Kalle Rosenbaum, Manning Publications.
2. http://rosenbaum.se/book/grokking-bitcoin.htmlModern compiler implementation in C
3. Blockchain Basics, Daniel Drescher, Apress Publication

List of Practicals:

LAB JOURNAL_ MCA676E	
S. No.	**TITLE**
1.	Study Fermat's and Euler's theorem.
2.	Study of RSA algorithm.
3.	Study of Diffie-Hellman Key Exchange Algorithm.
4.	Study of SHA 256 algorithm.
5.	Study of Digital Signature.
6.	Study of fundamental blockchain concepts, and block structure.
7.	Study of Genesis, Orphaned, and Uncle blocks.
8.	Study of Merkle Tree.
9.	Study of double spending problem in Blockchain.
10.	Case Study of BiCoin (BTC).
11.	Case study of Ethereum (ETH).
12.	Case study of Polkadot (DOT).
13.	Study of 51% rule and its implications.
14.	Study of consensus algorithms.
15.	Case study of Proof of Work (PoW) consensus algorithm.

Course Title:	Quantum Computing	**Course Code:**	MCA677E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA503C,MCA579E	**Co-Requisite:**	MCA676E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The course will provide an insight of basic of quantum physics from a computer scientist's kperspective, and how it describes reality and understand the philosophical implications of quantum computing

COURSE CONTENT:

UNIT I

Qubit & Quantum States: The Qubit, Vector Spaces. Linear Combination Of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, gram-schmidt orthogonalization, bra-ket formalism, the Cauchy-schwarez and triangle Inequalities.

UNIT II

Matrices & Operators: Observables, The Pauli Operators, Outer Products, The Closure Relation, Representation of operators using matrices, outer products & matrix representation, matrix representation of operators in two dimensional spaces, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators.

UNIT III

Commutator Algebra, Heisenberg uncertainty principle, polar decomposition & singular values, Postulates of Quantum Mechanics.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Achieve knowledge of Vector spaces, Matrices, Quantum state, Density operator and Quantum Measurement theory.

Text Books:

1. Quantum Computing without Magic by Zdzislaw Meglicki
2. Quantum Computing Explained By DAVID Mc MAHON

References:

1. Quantum Computer Science By Marco Lanzagorta, Jeffrey Uhlmann
1. An Introduction to Quantum Computing Phillip Kaye, Raymond Laflamme, Michele Mosca.

List of Practicals:

LAB JOURNAL_ MCA677E	
S. No.	**TITLE**
1.	**Introduction to Qubits & Vector Spaces** Represent qubits using the Dirac notation (bra-ket). Explore the properties of qubits in two-dimensional Hilbert space.
2.	**Linear Combination & Spanning Sets** Learn about linear combinations of vectors and spanning sets. Determine the uniqueness of a spanning set in a given vector space.
3.	**Basis & Dimensions** Identify bases and dimensions of qubit states. Explore the concept of orthonormality and apply the Gram-Schmidt orthogonalization process.
4.	**Inner Products & Triangle Inequalities** Calculate inner products of qubit states. Apply the Cauchy-Schwarz inequality and triangle inequalities to qubit states.
5.	**Observables & Pauli Operators** Learn about observables and their representation using Pauli operators. Explore the properties of Pauli operators in quantum mechanics.
6.	**Outer Products & Closure Relation** Calculate outer products of qubit states. Understand the closure relation in terms of outer products.
7.	**Matrix Representation of Operators** Learn how to represent operators using matrices. Convert outer products into matrix representations.
8.	**Eigenvalues & Eigenvectors** Calculate the eigenvalues and eigenvectors of qubit operators. Explore the concept of spectral decomposition in quantum mechanics.
9.	**Trace & Expectation Value** Calculate the trace of qubit operators and explore its properties. Compute expectation values of qubit operators.
10.	**Projection & Positive Operators** Understand projection operators and their properties. Explore the concept of positive operators in quantum mechanics.
11.	**Commutator Algebra & Heisenberg Uncertainty Principle** Learn about commutator algebra and its significance in quantum mechanics. Apply the Heisenberg uncertainty principle to qubit operators.
12.	**Polar Decomposition & Singular Values** Understand polar decomposition and singular value decomposition in quantum mechanics. Calculate singular values and explore their significance.
13.	**Postulates of Quantum Mechanics** Recap the postulates of quantum mechanics. Apply the postulates to solve simple quantum mechanics problems.

Course Title:	Advanced Java	**Course Code:**	MCA678E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA501C, MCA580E	**Co-Requisite:**	MCA657E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To learn about streams and how java is using functional programming concepts.
- To learn about database connectivity in java.
- To learn about socket programming in java.
- To learn about Spring Framework in java

COURSE CONTENT:

UNIT I

Streams: stream operations, stream creation, filter map & flatmap methods, extracting substreams & concatenating streams, other stream transformation, simple reductions, the optional type ,collecting results, collecting into maps, grouping & partitioning, downstream collections, reduction operations, primitive type streams, parallel streams Input & output i /o streams, text input & output, reading & writing binary data, object i/o streams & serialization.

UNIT II

Networking Connecting to a server, implementing servers, Database Programming JDBC design, structured query language, JDBC Configuration, Working with JDBC statements, query execution, scrollable & updatable result sets, row sets, metadata, transactions, advanced sql type, connection management in web & enterprise applications.

UNIT III

Core Spring: Introduction to spring, wiring beans, advanced wiring, aspect oriented spring, Spring on web building, spring web application , Spring in the backend using databases with Spring & JDBC, persisting database with object relational mapping, advanced spring MVC, working with spring web flow , securing web applications.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Interpret enumerations and collections in advanced Java.
- Build programs using collection framework.
- Illustrate and develop String Handling methods in JAVA.
- Apply Servlets to develop web applications.
- Demonstrate database access using JDBC API.
- Design reusable software components using JSP.

Text Books:

1. Core Java Volume 11-Advanced features by Cay S.horstman 10th edition Publisher: Prentice HallSpring in Action, Fourth Edition

References:

1. Pro Spring 5 Authors: Cosmina, I.Harrop, R., Schaefer, C.
2. Spring Boot in Action by Craig Walls

List of Practicals:

LAB JOURNAL_ MCA678E	
S. No.	**TITLE**
1.	Write a program to implement different stream operations in Java.
2.	Write a program to extract substreams from a given stream.
3.	Write a program to concatenate multiple stream and other stream transformations.
4.	Write a program to demonstrate input and output streams.
5.	Write a program to read and write binary data from/ to a file stream.
6.	WAP to connect database to Java program using JDBC and check the connection.
7.	Write a program to create a database table using Java
8.	Write a program to insert, update, delete & select records from a database table
9.	Write a simple "hello world" program using spring.
10.	WAP with spring in the backend using databases with Spring and JDBC

Course Title:	Deep Learning	**Course Code:**	MCA679E
Semester:	4th	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA612E, MCA552C,MCA602C,MCA604C	**Co-Requisite:**	MCA652C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- The course shall provide a comprehensive understanding of deep learning fundamentals, including neural networks, back propagation, and optimization algorithms.
- It will equip students with specialized knowledge in CNNs for image processing, RNNs/LSTMs for sequential data analysis, and practical skills using Tensor Flow and PyTorch.

COURSE CONTENT:

UNIT I

Introduction to Neural Networks: Understanding Perceptrons, Activation Functions - Sigmoid, ReLU, Tanh, Feedforward Neural Networks, Backpropagation, Algorithm, Loss Functions (e.g., Mean Squared Error, Cross-Entropy), Gradient Descent and Stochastic Gradient Descent, Regularization Techniques (e.g., Dropout, L2 Regularization), Optimization Algorithms (e.g., Adam, RMSprop), Model Evaluation and Metrics, Practical Implementation (e.g., using TensorFlow, PyTorch).

UNIT II

Convolutional Neural Networks (CNNs):Introduction to Image Processing with Neural Networks, Convolutional Layers, Pooling Layers (e.g., Max Pooling, Average Pooling), Padding and Stride, CNN Architectures (e.g., LeNet, AlexNet, VGG, ResNet), Transfer Learning and Fine-Tuning, Object Detection and Localization (e.g., YOLO, SSD), Image Segmentation (e.g., U-Net, FCN), Interpretability and Visualization Techniques, Advanced Topics (e.g., Attention Mechanisms in CNNs)

UNIT III

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM): Introduction to Sequential Data Processing, Basics of Recurrent Neural Networks (RNNs), Vanishing Gradient Problem in RNNs, Long Short-Term Memory (LSTM) Networks, Gated Recurrent Units (GRUs), Bidirectional RNNs Sequence-to-Sequence Models, Attention Mechanisms in RNNs, Applications in Natural Language Processing (NLP), Time Series Analysis and Forecasting with RNNs/LSTMs.

COURSE OUTCOMES:

At the end of the course student will be able to:

- Acquire a comprehensive understanding of deep learning concepts, including neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM).
- Gain proficiency in implementing deep learning models for image processing, natural language processing (NLP), and time series analysis using popular frameworks such as Tensor Flow and PyTorch.
- Develop the skills necessary to apply advanced deep learning techniques to real-world problems, including object detection, image segmentation, and sequence prediction, while also understanding interpretability and visualization methods.

Text Books:

1. Deep Learning/by Bengio Yoshua .,Courville Aaron.,Goodfellow Ian .Publisher Information : The MIT Press, London: 2016

References:

1. “Deep Learning” by Ian Goodfellow, Yoshua Bengio, and Aaron Courville:
2. “Neural Networks and Deep Learning: A Textbook” by Charu C. Aggarwal:
3. “Deep Learning for Computer Vision” by Rajalingappaa Shanmugamani:
4. “Python Deep Learning” by Ivan Vasilev and Daniel Slater:

List of Practicals:

LAB JOURNAL_ MCA679E	
S. No.	**TITLE**
1.	Image Classification using Convolutional Neural Networks (CNNs)
2.	Sentiment Analysis on Text Data using Recurrent Neural Networks (RNNs)
3.	Handwritten Digit Recognition with the MNIST dataset
4.	Object Detection using YOLO (You Only Look Once) algorithm
5.	Generative Adversarial Networks (GANs) for generating synthetic images
6.	Natural Language Processing (NLP) tasks like Named Entity Recognition (NER)
7.	Transfer Learning with pre-trained models like VGG, ResNet, etc.
8.	Reinforcement Learning for simple game environments like OpenAI Gym
9.	Time Series Forecasting using Long Short-Term Memory (LSTM) networks
10.	Autoencoders for dimensionality reduction or anomaly detection tasks.

NEP 2020 Based Focus Areas

Course Outline MCA Semester I

Course Code	Course Name	Course Type
MCA501C	Programming Concepts in C/C++	SEC
MCA502C	Database Management System	FC
MCA503C	Discrete Mathematics	FC
MCA504C	Computer Organization and Architecture	FC
MCA511E	Digital Electronics	SEC
MCA512E	Programming Languages & Paradigm	SEC
MCA513E	Management Information System	EC
MCA514E	Elements of Business Management	EOC
MCA526E	Web Technologies	EOC
MCA527E	E-Commerce	EC
MCA528E	Multimedia & Image Authoring	EC
MCA529E	Programming in R	SEC
MCA541A	Information and Communications Technology	AEC

Course Outline MCA Semester II

Course Code	Course Name	Course Type
MCA551C	Data Structures	FC
MCA552C	Artificial Intelligence	AEC
MCA553C	Software Engineering	FC
MCA554C	Operating System	FC
MCA561E	Advanced Computer Organization & Architecture	AEC
MCA562E	Operational Research	AEC
MCA563E	Microprocessor & Assembly Language Programming	AEC
MCA564E	Soft Computing	SEC
MCA576E	Modeling & Simulation	SEC
MCA577E	Computer Graphics	SEC
MCA578E	Numerical and Statistical Computing	VAC
MCA579E	Programming in Python	SEC
MCA580E	Java Programming	EC
MCA591A	Technical Communication	VAC

Course Outline MCA Semester III

Course Code	Course Name	Course Type
MCA601C	Full-Stack Development	EC
MCA602C	Machine Learning	AEC
MCA603C	Data Communication & Computer	FC
MCA604C	Analysis and Design of Algorithms	FC
MCA611E	Advanced Operating Systems	AEC
MCA612E	Distributed Databases	SEC
MCA613E	Bioinformatics	AEC
MCA614E	System Software Design	AEC
MCA615E	Natural Language Processing	AEC
MCA626E	Digital Image Processing	AEC
MCA627E	Linux System Programming	SEC
MCA628E	Data Warehousing	EC
MCA629E	Open Source Technologies	EOC

Course Outline MCA Semester IV

Course Code	Course Name	Course Type
MCA651C	Theory of Formal Languages	FC
MCA652C	Big Data Analytics	AEC
MCA653C	Project work	EOC
MCA661E	Wireless Communications	SEC
MCA662E	Cloud and Grid Computing	SEC
MCA663E	Information Security and Networks	SEC
MCA664E	Compiler Design	AEC
MCA676E	Block Chain Technologies	EC
MCA677E	Quantum Computing	AEC
MCA678E	Advanced Java	EC
MCA679E	Deep Learning	SEC

Course Summary

Foundation Courses	9
Skill Enhancement Courses	15
Value Added Courses	2
Ability Enhancement Courses	14
Employability Courses	8
Entrepreneurship Oriented Courses	4