Course: MBA **Semester:** 1

Prerequisite: Proficiency in English language and familiarity with fundamental concepts of functional English grammar.

Rationale : This course builds professional communication and employability skills, bridging academic learning with workplace expectations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			Marks EYE		
					T	CE	P	T	P	
1	-	2	-	2	40	-	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Self-Introduction – SWOT & SOAR, Body Language and Professionalism**Purpose of Self-Introduction (Academic vs Professional) Structure of an Effective Self-Introduction Understanding SWOT & SOAR (Concept and Use) Professional Appearance and Grooming Basics of Positive Body Language	10	1
2	**Sentence correction, One-word substitution, Para jumbles, Cloze Passages, Vocabulary-Synonyms, Antonyms**Common Grammar Rules for Sentence Correction Parts of Speech & Usage Logical Flow in Paragraphs Contextual Vocabulary Learning Synonyms & Antonyms – Word Relationships	10	2
3	**Writing Skills: Paragraph Writing, Letter Writing - Formal, Email Writing**Paragraph Writing – Structure & Types Letter & Email Writing – Formal Formats & Do's/Don'ts	10	2
4	**Reading Comprehension (Level of Difficulty - Intermediate) – Tips and Practice**Types of Reading (Skimming, Scanning, Intensive) Identifying Main Idea & Supporting Details Inference and Vocabulary from Context Time Management in Reading Tests	10	1
5	**Extempore**Meaning and Importance of Extempore Structure of a Short Speech Handling Stage Fear Time Management Techniques	10	1
6	**Goal Setting**Meaning and Importance of Goals Short-term vs Long-term Goals SMARTER Goal Concept Personal vs Professional Goals	10	1
7	**Team Building and Convincing Skills**Meaning of Teamwork Roles in a Team Qualities of a Good Team Member Basics of Persuasion & Convincing	10	1

	Ethical Convincing vs Manipulation		
8	**Negotiation Skills**Meaning and Importance of Negotiation Types of Negotiation Win-Win vs Win-Lose Approach Negotiation Stages Communication in Negotiation	**10**	**2**
9	**Presentation Skills - Do's and Don'ts - Making effective Presentations**Purpose of Presentations Structure: Introduction–Body Conclusion Do's and Don'ts of Slides Common Presentation Mistakes Visual Aids & Audience Engagement	**10**	**2**
10	**Debate**Format of a Debate Rules and Roles (Proposition/Opposition) Argument Building & Rebuttal Logical Reasoning in Debate	**10**	**2**
	Total	**100**	**15**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Analyze	Evaluate
Weightage	10	15	55	20

Reference Books

1.	**Business Communication Today**
2.	**Objective General English. S. Chand,** By Aggarwal, R. S.
3.	**English Grammar in Use** By Murphy, Raymond | Cambridge University Press, Pub. Year 2019
4.	**English Grammar in Use** By Raymond Murphy
5.	**Soft Skills: Key to Success in Workplace and Life.** By Meenakshi Raman
6.	**Basic Communication** By Poe & Fruchling | AITBS
7.	**The 7 Habits of Highly Effective People** By Stephen R. Covey | Free Press
8.	**Personality Development and Soft Skills** By Barun K Mitra | Oxford, 2011
9.	**Communication Skills. Oxford University Press, 2017.** By Raman, Meenakshi, and Sangeeta Sharma.
10.	**Basics of Business Communication** Cambridge University Press, | South Asian Edition

Course Outcomes

At the end of this course Students Will be able to:	
1	Remember key concepts related to self-introduction, SWOT/SOAR analysis, body language, professionalism, and basic vocabulary including synonyms, antonyms, and one-word substitutions.
2	Understand sentence structure and contextual meaning by interpreting para-jumbles, cloze passages, vocabulary usage, and intermediate-level reading comprehension passages.
3	Analyse effective communication techniques through paragraph writing, formal letter writing, email drafting, extempore speaking, debates, and structured self introductions.
4	Analyse communication situations to enhance presentation effectiveness by identifying do's and don'ts, evaluating body language, team dynamics, convincing strategies, and negotiation approaches
5	Evaluate personal communication performance by assessing goal-setting strategies, teamwork contributions, negotiation effectiveness, and overall professionalism for continuous improvement.

List of Practical

1.	**Self-Introduction – SWOT & SOAR, Body Language and Professionalism** 60-second Self-Introduction Practice SWOT/SOAR Worksheet Activity Posture, Eye Contact & Hand Movement Drill Peer Feedback on Professional Presence
2.	**Sentence Correction, One-word substitution, Para jumbles, Cloze Passages, Vocabulary-Synonyms, Antonyms** Sentence Correction Worksheets One-word Substitution Quiz Para-jumble Rearrangement Game Cloze Passage Practice Sets Vocabulary Building through Word Games Daily Word Challenge Activity
3.	**Writing Skills: Paragraph Writing, Letter Writing - Formal, Email Writing** Write a Paragraph from a Given Topic Write a Leave/Application Letter Draft a Professional Email Identify Errors in paragraph, letter and email
4.	**Reading Comprehension (Level of Difficulty - Intermediate) – Tips and Pract** Timed Reading Comprehension Practice Question-type Analysis
5.	**Extempore** Pick-and-Speak Activity 1-minute Extempore Rounds
6.	**Goal Setting** Personal Goal-Setting Worksheet SMARTER Goal Conversion Activity
7.	**Team Building and Convincing Skills** Team Task Challenge Problem Solving Group Activity Convince the Group Exercise
8.	**Negotiation Skills** Role-play: Buyer–Seller Negotiation Win-Win vs Win-Lose Approach Salary Negotiation Simulation
9.	**Presentation Skills - Do's and Don'ts - Making effective Presentations** Presentation Skills – Slide Preparation & Tips Create a Short Presentation Individual Presentation
10.	**Debate** Classroom Debate Sessions

Course: MCA **Semester:** 1

Prerequisite: Understanding of Object Oriented Design, UML

Rationale : To analyze the requirements of end users, Understands the software development life cycle (SDLC) along with many other process models, gain understanding of test strategies and different types of testing, gain understanding of different agile processes

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	2	-	4	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to System and Software Development** Concept of system, Basic components of system, Information systems categories, Need of information system development. Study of different models, Software characteristics, Components, Applications, Layered technologies, Processes, Methods and tools, Generic view of software engineering, Classical Systems Development Life Cycle (SDLC) method.	15	6
2	**Requirements Analysis & Engineering** **Requirement Analysis:** Classical Systems Development Life Cycle (SDLC) method, Requirement determination, System Requirement Specification (SRS), Fact finding techniques. Process models - waterfall model, incremental, Evolutionary process models - prototype, spiral and concurrent development model. **Requirement Engineering:** Problem recognition, Requirement engineering tasks, , Use cases and functional specification, Requirements validation	20	10
3	**Structured System Analysis & Design** **System Analysis:** Introduction to Structured System Analysis Development Methodology (SSADM), Tools for analysis - decision trees, decision tables, structured english, Data flow diagram, Entity Relationship (ER) diagram, Data dictionary. System Design: Design concepts, Design modeling, Software architecture, Data design, Architectural styles and patterns, Procedural design, Object oriented design.	20	10
4	**Agile Methodology** Agile process, Extreme Programming (XP), Brief overview of other agile process models - adaptive software development, Scrum.	20	7
5	**User Interface Design** Concepts of UI, Interface design model, Internal and external design, Evaluation, Interaction and information display software.	5	3
6	**Planning a Software Project** Management spectrum, People, Product, Process, Project, W5HHPrinciple, Importance of team management, Scope and feasibility, Effort estimation, Schedule and staffing, Quality planning, Risk management- identification, assessment, control,	15	6
7	**Case Tools and Study** Introduction to CASE, Building blocks of CASE, Integrated CASE environment.	5	3
	Total	100	45

Reference Books

1.	**Software Engineering : A Practitioner's Approach (TextBook)** By Pressman R.S | TMH
2.	**Software Engineering** By Sommerville
3.	**Software Engineering** By Rajiv Mall | PHI
4.	**Software Engineering** By Pankaj Jalote | Wiley India
5.	**Software Engineering, An Engineering Approach** By Peters & Pedrycz | Wile-India
6.	**Software Engineering, Principles and Practice** By Jawadekar | TMH

Course Outcome

After Learning the Course the students shall be able to:

1. describe Software Development Life Cycle. explain software development process in association with its fundamental principles and methodologies.
2. analyze & represent end user requirements and model requirement analysis using Unified Modeling Language.
3. prepare & represent software design and design software model using Unified Modeling Language.
4. define significance of project planning, effort estimation and risk management. design test cases and identify testing strategies.

List of Practical

1.	**Analyze a Library Management System and identify its system components — input, process, output, feedback, control, boundary, and environment — and document them in tabular form.**
2.	**Examine a College Examination System and identify the type(s) of Information System (TPS, MIS, DSS, EIS) it uses, with proper justification.**
3.	**Apply the Classical Systems Development Life Cycle (SDLC) by mapping each phase to the problem of an Online Admission System in correct sequence.**
4.	**Conduct requirement elicitation for an Online Voting System by performing one interview and one questionnaire, and document the elicited requirements.**
5.	**Classify and document exactly five functional and five non-functional requirements for an Online Examination System .**
6.	**Prepare a System Requirement Specification (SRS) document for a Library Management System including a minimum of five standard SRS sections.**
7.	**Design a Context Diagram, Level-0 DFD, and one Level-1 DFD for a Student Information System , ensuring proper balancing between levels.**
8.	**Develop an Entity-Relationship (ER) Diagram with at least four entities for a Banking System , clearly indicating attributes and relationships.**
9.	**Formulate exactly five user stories for an Online Food Delivery Application using standard Agile user-story format.**
10.	**Prepare a Software Project Plan for a Mini Software Project including project scope, effort estimation, schedule, and identification of three major risks.**

Course: MCA **Semester:** 1

Prerequisite: Knowledge of DBMS

Rationale : This course is intended to give students advanced concepts of Relational Database Management System, security aspects of databases and introduction to other databases. Also concepts and practical aspects of data manipulation using stored procedures and triggers will be given

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	2	-	4	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to Databases** **Basic Concepts:**Data, Database, Database systems, DBMS, Purpose and advantages of DBMS. **Data Models:**Introduction, Three level architecture, Various components of a DBMS. **Overview:** Parallel database, Distributed database, Object oriented database, Object Relational Database, Comparison of RDBMS, OODBMS and ORDBMS.	10	5
2	**Relational Data Model and Database Design** **Relational Structure:** Tables, Rows, Columns, Entity sets, Attributes, Types of entities, Relationships and types of relationships, Database modelling using entity and relationships. **Keys:**Super key, Candidate keys, Primary key, Entity integrity constraints, Referential integrity constraints. **Indexing:** Types of single level ordered indexes, Primary index, Cluster index, Secondary index, and Multilevel index. **DB Design:** Database design process, Functional dependencies, Normalization.	15	8
3	**Query Language and Database operations** Overview of SQL, Basic and Advanced queries in SQL, Aggregation, Views in SQL, Introduction of NoSQL, Relational Algebra and Calculus – basic and advanced operators**.**	10	6
4	**PL/SQL, Cursor and Trigger** Basic code structure, Variables, Conditional statements, Looping Structures, Cursor Operations, Triggers.	15	7
5	**Stored Procedures** Understanding the main features of stored procedures, stored procedure architecture, Advantages of using procedures. Stored procedures - functions, procedures and packages.	20	8
6	**Database Transactions and Database Security** Overview of Database Transactions, concurrency control, Deadlock Handling. Database security and its issues, Granting and Revoking privileges, Role based access control for multilevel security, Encryption and PKI, Challenges in database security.	20	7

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
7	**Backup and Recovery in Database** Database Backup and Recovery concepts, Database Backup and Recovery techniques, Current trends of database technologies.	10	4
	Total	100	45

Reference Books

1.	**Database System Concepts** By Silberschatz, Korth, Sudarshan | McGraw Hill Publication | 4th Edition
2.	**SQL, PL/SQL – The Programming Language** By Ivan Bayross | BPB Publications
3.	**Database Management Systems** By Raghu Ramakrishnan, Johannes Gehrke | McGraw Hill Publication
4.	**Fundamentals of Database Systems** By Ramesh Elmasari, Shamkant B. Navathe | Pearson Education

Course Outcome

After Learning the Course the students shall be able to:

1. identify significance of object relational database management system.
2. apply programming construct such as stored procedure, stored function, cursor and triggers.
3. define different types of index and database security mechanism.
4. discuss emerging database technology such as mobile database, multimedia database and geographical information system database.

List of Practical

1.	**Stored Procedure** With Table a) To display content of product table(pid, pname, pamt)out Using Table b) To display message “Welcome to mysql” Parameter (IN OUT) c) To calculate simple interest, display it store interest in file administrator d) To display name of the employee whose id is given by you e) To add two numbers Parameter(INOUT) f) BY setting different values of counter
2.	**Stored Functions** 1. Stored function that accept no parameter: Write stored function to find out total number of employees in the organisation 2. Stored function that accepts the parameter: Write stored function that will change the name of the employee whose emp_id is 005 from employee 3. Write stored function to find out average marks scored by student from student table
3.	**Stored functions Assignment** 1. Write stored function to update a record 2. Write stored function to return customer level such as PLATINUM, GOLD, SILVER based on customer credit limit. Customer(cust_id, Cust_name, Cust_creditlimiterage product price from product table
4.	**Loop statement** 1. Write stored procedure to display 1 to 10 numbers using loop statement 2. Write procedure to display 1 to 5 numbers in reverse order

	3. Write procedure to find out the final value of x , which is initialize to zero and incremented by 1 till it reach to value key in by user
5.	**Loop Assignment** 1. Write Function to compare two numbers and display any one of the following string depending upon the resulta n < m n > m n = m b. Write Function to compare two numbers and display any one of the following string depending upon the result in verbose manner n less than m n greater than m n equals m
6.	**Procedure using while loop**
7.	**Cursors** 1. Stored procedure that builds an email list of all employees 2. Write cursor to retrieve recordes from test111 table. Test111 consist of ID and name 3. Write stored procedure to creates a new table on the fly (if it does not exist) named ordertotals. This table will store the results generated by the stored procedure. And display table
8.	**Cursor Assignment** Cursor Assignment a) Supplier (sid, sname, contactnum) Parts (pid, pname, color, unit rate) Catalog (sid, pid, qty) 1. Implement the following: i. Find the top three Parts been ordered and have the highest sale till date. ii. Find those suppliers who charge more for some part that the average cost of that part. 2. Implement the following: 1. Create block to prepare invoice in following format and display it. Use parameterized constructor 2. Create a block to prepare invoice in following format. 3. Prepare report for Part Use parameterized cursor. _____ Part Id Part Name Quantity (in Hand) Unit Price Total Parts Available: b) Creatte a program to display all the records from table student using cursor. (ASSIGNMENT) c)Create table Patient (Patient_code, Patient_name, Address, City, DOB, Gender) Create block USING CURSORS to generate the following report: Gender (Patients in Age group) 1-20 21-40 41-60 61-80 81-100 Total _____ Male _____ Female Total
9.	**Triggers** 1. Create trigger before insert to sum the values inserted 2. Write trigger before update last name of the employee as “Phan” whose employee number is 1056 3. Write trigger after insert to display newly inserted record Table contacts (contact_id INT(11) NOT NULL AUTO_INCREMENT, last_name VARCHAR(30) NOT NULL, first_name VARCHAR(25), birthday DATE, created_date DATE, created_by VARCHAR(30), Audit table consist of

	contact_audit(contact_id, Inserted_date, inserted_by) d. Write trigger to maintain Blog application. Following two tables are required: `blog`: stores a unique post ID, the title, content, and a deleted flag. `audit`: stores a basic set of historical changes with a record ID, the blog post ID, the change type (NEW, EDIT or DELETE) and the date/time of that change
10.	**Index and Security** 1. Index 2. Create index officecode on employee table 3. Remove index officecode on employee table 2. Security 1. Create a super account that can do anything including being able to grant privileges to other users 2. To create a user that has all access in the sample database and can connect from any host 3. Revoke the privileges

Course: MCA **Semester:** 2

Prerequisite: Knowledge of programming language (C/C++).

Rationale : To provide understanding of basics of data structures and their operation for efficient storage and retrieval of data.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/	Credit	T	CE	P	T	P	
4	0	2	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction** Primitive and non-primitive data structures, String manipulation and pattern matching, Storage representation of strings, Text handling, Key Word In Context (KWIC) indexing, Arrays, Storage structure for arrays, Special types of arrays – triangular and sparse.	10	7
2	**Stack and Queue** Stack, Stack operations, Applications of stack - recursion, polish notations - prefix, infix, postfix, Algorithms of stack applications, Introduction to queue, Algorithms and implementation of simple queue, Circular queue, Double ended queue, Priority queue.	20	12
3	**Linked List** Linked list, Algorithms and implementation of singly linked list, Doubly linked list, Circular linked list, Operations on linked list, Applications – polynomial representation, addition of two polynomials.	20	8
4	**Trees** Concept and terminologies of tree, General tree, Binary tree and its storage representation, Binary search tree and its operations – create, insert, delete, Traversal of tree - inorder, preorder, postorder, Threaded tree, B tree and B+ tree, Height balanced tree - AVL tree, Rotations in AVL tree, Applications – heap tree, expression tree.	20	12
5	**Graph** Concept and terminologies of graph, Representation of graph -adjacency matrix, adjacency lists, Introduction to graph traversal - Depth First Search (DFS), Breadth First Search (BFS), Introduction to spanning tree.	10	5
6	**Searching, Sorting and Hashing** Linear search, Binary search, Bubble sort, Selection sort, Insertion sort, Shell sort, Quick sort, Heap sort, Merge sort, Radix sort, Hashing, Hashing functions, Collision resolution techniques.	20	13
	Total	100	57

Reference Books

1.	**An Introduction to Data Structures with Applications (TextBook)** By Jean-Paul Tremblay, Paul G. Sorenson | Tata McGraw-Hill | 2nd Edition, (2007)
2.	**Introduction to Algorithm** By Cormen, Leiserson, Rivest, Stein | PHI (2003) | 2nd Edition
3.	**Data Structures using C and C++** By Tanenbaum | PHI
4.	**Expert Data Structures with C** By R. B. Patel
5.	**Theory and Problems of Data Structures** By Seymour Lipschutz | Schaum's Outline Series
6.	**Data Structures Through C++** By Yashavant Kanetkar | BPB

Course Outcome

After Learning the Course the students shall be able to:

1. describe the significance of various linear and non-linear data structures such as arrays, stack, queue, linked list, trees and graph.
2. identify the appropriate data structure for a given problem.
3. construct most suitable data structure to solve a problem by considering various problem characteristics such as data size and various type of operations.
4. design and implement various techniques for searching, sorting and hashing.

List of Practical

1.	**Write a program to perform various stack operations using array**
2.	**Write a program to convert infix expression to prefix and postfix expression using stack**
3.	**Write a program to perform insert and remove operations on following** a. Simple Queue b. Circular Queue c. Priority Queue
4.	**Write a program to perform Double Ended Queue [Input Restricted / Output Restricted]**
5.	**Write a program to create a singly link list in FIFO & LIFO form**
6.	**Write a program to perform following singly link list operations** a. insert b. delete c. search d. reverse
7.	**Write a program to create a doubly link list in FIFO & LIFO form**
8.	**Write a program to perform following doubly link list operations** a. insert b. delete c. search d. reverse
9.	**Write a program to add two polynomials**
10.	**Write a program to perform following circular link list operations**

	a. insert b. delete
11.	**Write a program to create a binary search tree and print its element in** a. Inorder b. Preorder c. Postorder
12.	**Write a program for insertion of a node in B tree / B+ tree**
13.	**Write a program to create a graph in a adjacency list structure traverse it in** a. DFS b. BFS
14.	**Write a program to perform following sort** a. Bubble Sort b. Selection Sort c. Insertion Sort d. Shell Sort e. Quick Sort f. Heap Sort g. Merge Sort h. Radix Sort
15.	**Write a program to search an element using** a. Linear Search b. Binary Search

Course: MCA **Semester:** 3

Prerequisite: Knowledge of C, C++

Rationale : To acquire the fundamental knowledge of Java programming

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	2	-	4	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to Java Data Types, Operators, Statements** Paradigms of programming languages, Evolution of OO methodology, Basic concepts of OO approach, Comparison of object oriented and procedure oriented approaches. Concepts of OOP - classes and objects, abstraction and encapsulation, inheritance, polymorphism. Features of the Java language, Java environment, Object oriented programming in Java, Java program structure, Java and unicode, Data types, Variables and arrays - data types in Java, literals, characters, variable declaration, symbolic constants. Type casting operations in Java, Arithmetic operators, Basic assignment operators, Relational operators, Boolean logical operators, Ternary operator, Operator precedence. Control statements - Java's selection statements, switch, nested switch, iteration constructs, continue, return.	20	8
2	**Inheritance, Sub Classing, Package** Inheritance Concepts – defining sub classes, method overriding, using super keyword, Variable shadowing, Method and variable binding, Using final keyword, Abstract classes and interfaces. Object class, Packages - Creating package, CLASSPATH environment variable, access specifiers, Access Control / visibility.	20	8
3	**Exception, Collection Frameworks** Exception handling - types of exceptions, Throwable class, Keywords - try, catch, throw, throws and finally, Nested try statements, Java built in exceptions, User defined exceptions. Collection framework – Collections, List, Set, Enumeration, Iterator, ArrayList.	15	8
4	**IO and Multi Threading** java.io- File class, Creating directory, Input/Output basics, Streams (byte and character), Reading from and writing to console, Reading and writing files, PrintWriter Class, the transient modifier, RandomAccessFile, Introduction to multi-threading, Thread class and execution of thread, Runnable interface, ThreadGroup, Daemon threads, Thread states.	15	8
5	**GUI Programming and Applets** Introduction to AWT and Swing, Fundamentals of applets, Applet class, Applet life cycle, A simple banner applet, getDocumentBase(), getCodeBase(), showDocument(), AppletContext and AppletStub interface, Working with frames, windows, graphics, color, fonts. AWT controls - buttons, checkbox, choice, list and textField. Layout Managers - Flow Layout, Grid Layout and Border Layout. User interface events - event classes and event listener interfaces, Adapter classes.	30	13
	Total	100	45

Reference Books

1.	**The Complete Reference Java J2SE (TextBook)** By Herbert Schildt | TMH Publishing Company Ltd | 5th Edition
2.	**Core Java Volume 1** By Cay Horstmann and Gary Cornell | Pearson Education | 8th Edition
3.	**The class of JAVA** By Pravin Jain | Pearson

Course Outcome

After Learning the Course the students shall be able to:
1. implement object oriented principles using Java. 2. identify errors and implement exception handling mechanism. 3. develop programs using multi-threading concepts. 4. design and develop java applications using Applets, Abstract Window Toolkit and Swing API.

List of Practical

1.	**Basic datatype and looping** Write a program for swapping and find a factorial value. Perform swapping without using third variable.
2.	**Looping and Control structure** Write a program to accept a number from the user through command line and display whether the given number is palindrome or not.
3.	**Array** Write a program to accept an array of integer from the user through command line and find prime numbers from the array.
4.	**Class** Create a class Stack that defines an integer stack that can hold 10 values. Perform push and pop actions in a stack.
5.	**Inheritance** Write a program to create a class Publisher with attributes publisher name and publisher id. Derive a subclass Book with attributes bookname, bookid and author name. All these data should be entered by the user. Create two methods getdata() and showdata() to display the details of book and publisher.
6.	**Method Overloading** Write a program to create a class with two methods with same name addfunc(), one accepting two integer parameters and other accepting two double parameters. When method is called, the appropriate method should be selected depending on parameters passed(method overloading).
7.	**Super and this keyword** Declare a variable called x with integer as the data type in base class and subclass. Make a method named as show() which displays the value of x in the superclass and subclass.

8.	**Final class, abstract class and interface** Write a program to calculate the area, circumference and volume for all shapes. [Perform this application using final class, abstract class and interface]
9.	**Exception handling** Write a program to enter two integers from the command line and display the division of those two numbers. Handle all the exceptions (i.e. ArrayIndexOutOfBoundsException, NumberFormatException, ArithmeticException) for invalid arguments passed.
10.	**File Handling** Write a program to perform following actions and store output in file: 1. Accept strings from user, convert it into uppercase and store it in a file. 2. Write double value to a text file and also display the date on which the application was run inside the file. 3. Delete a given file or directory and display appropriate message
11.	**Collection Framework** Write a program for creating an ArrayList, add the elements in array list and then obtain an array from ArrayList and display the contents and sum of those numbers.
12.	**Collection Framework** Create a class named Address and define name, city and state as the data members of this class. Create another class named Maillist and add the elements to the linked list and display the contents using Iterator interface.
13.	**Calendar class** Create a Gregorian Calendar. Display current date and time in default locale and time zone. Find out whether the current year is a leap year or not.
14.	**Regular expression** Write a program to find integers and decimal value from the string by using a pattern of regular expression
15.	**Thread and Runnable interface** Write a multi-threaded program which sets the priority of threads and gets the name of threads.
16.	**Thread (sleep method)** Write a program to calculate sum and factorial of numbers using sleep method.
17.	**Multi-threading** Write a program to create multi-threaded application to perform banking tasks. [For Example: Withdrawal and Deposit from Joint account.]
18.	**AWT and Swing** Write a program to create a Menubar and create 2 Menus File and Edit. Involve New, Open, Close as items in the File menu

	and then add a separator, then further add Save, Save As and again add separator and add another menu item named Print. Add all these items in File menu. Add Line, Rectangle and Circle as the menu items and add Radio Button before these menu items and add a separator. Then add Red, Green and Blue as menu items and add Check Box before these menu items.
19.	**Applet and AWT** Create Java Applets to perform following tasks: 1. To display simple calculator 2. To write the content of the text area in the file, whose name is given in the text box using Frame 3. To draw rectangle to the applet when mouse is dragged.
20.	**AWT and Swing** Write a program to create 3 radio buttons named C++, Java and Pascal and add on JPanel. The user is asked to identify which of them is not an OOP language. When the user selects on choice, the program responds by displaying whether the selection is correct or wrong.

Miscellaneous

Useful Links

https://sites.google.com/a/paruluniversity.ac.in/java_div1/

Course: MCA **Semester:** 1

Prerequisite: Working knowledge of HTML, JavaScript, CSS and database concepts.

Rationale : To be able to develop web application using open source technologies, PHP scripting language and deploying application on Apache Web Server, Apache Web Server configuration and MySQL database deployment for web applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	4	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to open source Softwares and their Configuration**Overview of Open Source Softwares, Widely used open source products, Development philosophy, Open source vs. closed source, Open source software vs. free software, Open source vs. source available, Pros and cons, Development tools, The distribution terms of open source software, open source technology importance, Free and Open Source Software (FOSS), Configuring apache, Configuring Mysql, Configuring PHP	15	9
2	**Overview of PHP structure and syntax**Background information of PHP, Using variables, operators and expressions. Conditional statements and iterations in PHP: Conditional Statements: if statement, switch statement. Looping: for loop, while loop, do…while statement, for each statement. Functions and Arrays in PHP: PHP functions, creating array. PHP image manipulation.	20	10
3	**Accessing Mysql with PHP** Learning the MySQL data types, Frequently used string functions in MySQL, Using date and time functions in MySQL, Using transactions and stored procedures in MySQL, Interacting with MySQL using PHP, MySQL versus MySQLi functions, Connecting to MySQL with PHP, Working with MySQL Data.	25	10
4	**Working with User Data and Error Handling** Handling HTML forms with PHP, Sessions, Cookies, Handling and avoiding errors, Exception handling, Using AJAX with PHP.	20	10
5	**Object Oriented Programming using PHP** Creating classes and objects in PHP,Working with methods,Inheritance,Constructors,Destructors,Self and parent keyword Object cloning,Object Oriented Programming using MySQL DB.	20	9
	Total	100	48

Reference Books

1.	**Professional LAMP: Linux, Apache, MySQL and PHP5 Web Development** By Jason Gerner, Morgan Owens, Elizabeth Naramore, Matt Warden | WROX publication
2.	**Sams Teach Yourself PHP, MySQL and Apache All in One (TextBook)** By Julie C Meloni | Pearson Education
3.	**Open source web development with LAMP** By James Lee and Brent Ware | Pearson Education

Course Outcome

After Learning the Course the students shall be able to:
1.Understand the fundamentals of open source software, its configuration, and working with tools like Apache, MySQL, and PHP. 2.Apply PHP syntax, control structures, functions, and arrays to develop dynamic web applications. 3.Analyze and implement database operations by integrating MySQL with PHP for data handling and transactions. 4.Develop web applications using user data handling, error management, AJAX, and object-oriented programming concepts in PHP.

List of Practical

1.	**HTML Page Design** Design your Profile page with following details: - Personal Details - Educational Details - Profile Picture - Link of your workspace directory
2.	**Web Page design with CSS** Create a general stylesheet to apply on your developed webpages. [Use CSS class, CSS Groups and CSS tags]
3.	**Registration Form Web** Develop an online Admission Registration page.
4.	**Type Conversion through Script** Write a script to perform all possible type casting and conversion operations in PHP. Design output page in following format. Variable Name Variable Type Converted Type Converted Value
5.	**Webpage Development "Course Details"** Develop a webpage for “Course Details”. Display all course names with different colors.
6.	**String Operations** Write a script to perform all possible string operations on given inputted string values. Design output page in following format. String value1 String Value2 Operation Result
7.	**Login and SignUp Page** Display Login and SignUp page and redirect to Profile or Registration page accordingly.
8.	**Student Result Generation System** Develop a Student Result Generation System. Provide following facilities: - Students Exam Detail Entry - Marks entry - Result Generation - Marksheet and Result Analysis
9.	**Date and Time Functions** Write a PHP script to demonstrate Date and Time functions. Display given date/time, operation and result.
10.	**Server-side Validations using regular Expression** Design online placement registration form. Check for the validity of entered fields using regular expressions.
11.	**Sending a Mail** Develop a php script to send a mail. [For Example: Forgot password recovery mail.]
12.	**Menu Driven Program** Write a menu driven program to perform various file operations. - Display size of file - Display Last Access, changed, modified time of file - Display details about owner and user of File - Display type of file - Delete a file - Copy a file - Traverse a directory in hierarchy - Remove a directory
13.	**Image Manipulation**

	Develop a web application to perform PHP image manipulation. Perform following tasks: - Load image - Resize image - Crop image - Merge (watermark) - Output image to browser with format conversion.
14.	**Simple Image Gallery** Write a php scipt to design a simple gallery.
15.	**Error and Exception Handling** Develop a web application to perform Error and Exception Handling Operations. Implement proper requirements for demonstration.
16.	**Dynamic Search Box using AJAX** Develop a dynamic search box to find records using php and AJAX. [For Example: Google Search]
17.	**Event Webpage [Object Oriented Concepts]** Develop an online Event webpage. Provide admin to set font-size, color and event label for each event. Create Event class with constructor/Destructor. Use show_event() method to display event-list.
18.	**Object Oriented PHP Web Application** Create a web forum application that has a Member class for forum members containing methods such as createPost(), editProfile(), showProfile(). Create a class called Administrator that is a child of the Member class by adding extra methods such as createForum(), deleteForm() and banMember(). Override login() or logout method of parent class in child class.
19.	**PHP MySQL Database Connectivity** Create a skeleton of SubjectMIS class and define and implement methods to perform following MySQL functions on stored subject details: - Select, Insert, Delete, Update, Connect, Disconnect
20.	**Cricket Application using Interface** Define an interface for Cricket Game Activities. Implement this interface in a class. Write necessary class and member function definitions for a cricket player object. The program should accept details from user (player_code, name, runs, innings_played, no_of_times_out). The program should contain following menu. a) Enter details of players. b) Display average runs of a single player. c) Average runs of all players. d) Display the list of players in sorted order as per runs

Miscellaneous

Useful Links

https://sites.google.com/a/paruluniversity.ac.in/php/
http://www.php.net/

Course: MCA **Semester:** 2

Prerequisite: Knowledge of Core Java Programming and HTML.

Rationale : To make the students aware of the data driven web applications, web services, MVC architecture and ORM concepts of modern web application development.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
4	0	4	-	6	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Java Database Connectivity**Introduction to JDBC, JDBC architecture. Common JDBC Components - DriverManager, Driver, Connection, Statement, ResultSet, SQLException. Types of JDBC drivers, Establishing connection with database, Submitting queries and obtaining results, ResultSetMetadata Interface, Precompiled statements, CallableStatement, JDBC transactions.	15	9
2	**Java Servlets**Configuring Java Development Environment, Downloading and configuring Apache Tomcat web server. Introduction to servlets, Servlet life cycle, Exploring servlet API, Basic servlet structure, Handling client requests, Generating response, Working with cookies, Session tracking. Servlet filters - Introduction to filters, creating custom filters, mapping filters to servlets, mapping filters to specific URL pattern.	20	11
3	**Java Server Pages**Introduction to JSP, Need of JSP, Advantages of JSP over servlet, Life cycle of a JSP page, JSP basic tags: declaration tag, scriptlet tag, expression tag. JSP implicit objects - out, request, response, application, session, pageContext, page, exception. Introduction to java beans, Reading and writing properties of JavaBeans in JSP pages. JSP 2.0 expression language - Use of EL, Invoking Expression Language, Preventing Expression language evaluation, Accessing scoped variables. Custom tag libraries - tag handlers, tag library descriptor. JSP standard tag library (JSTL): Downloading and installing JSTL in Java web application, Core tags (c:out, c:forEach, c:if, c:choose, c:set, c:remove, c:import, c:catch).	25	14
4	**Java Web Services**Introduction to web services, Architecture of web services, Types of web services - SOAP web services, REST web services SOAP v/s REST, Introduction to REST, Configuring RESTful web service using Jersey framework in java web application, RESTful architectural principles, HTTP method and URI matching. JAX-RS injection - PathParam, QueryParam, FormParam. Introduction to Server responses and exception handling, JAX-RS client API.	20	13
5	**Introduction to Spring and Hibernate**Introduction to Spring MVC Framework, Spring MVC Architecture and Components, Advantages of Spring MVC, Modules Of Spring Framework. Spring Application Introduction to Hibernate, Architecture of hibernate, Hibernate O/R mapping, Configuring hibernate development environment, Implementing hibernate O/R Mapping, Introduction to Hibernate Query Language (HQL).	20	13
	Total	100	60

Reference Books

1.	**Java Server Programming Java EE6 Black Book (TextBook)** Dreamtech Press
2.	**RESTful Java with JAX-RS 2.0** By Bill Burke | O'Reilly | 2nd Edition
3.	**Core Servlets and Java Server Pages Volume-1** By Mary Hall and Larry Brown | Prentice Hall | 2nd Edition
4.	**Core Servlets and Java Server Pages Volume-2** By Marty Hall, Larry Brown and Yaakov Chaikin | Prentice Hall | 2nd Edition

Course Outcome

After Learning the Course the students shall be able to:
1. design and develop data driven web applications using JDBC, servlet and JSP API. 2. develop MVC based web applications using Struts. 3. develop and integrate web services and web clients. 4. describe significance of Object Relational Mapping (ORM) in Java context.

List of Practical

1.	**JDBC Connectivity** Develop a java application using concept of JDBC for user login. User will enter username and password. Application will match for the same in database table. If match is found then display message "Successful Login" and if not found then display message "Invalid Username and password ".
2.	**JDBC Connectivity & CRUD Operations** Develop a Menu driven java application for student information which will create the table with appropriate columns. Menu will contain the options like insert, update and delete. Based on the option, data will be inserted or updated or deleted from table based on student id (student_id will be primary key). Display appropriate message for each operation.
3.	**Data Driven GUI Application** Develop an application using GUI for user registration. In first form user will enter the data for registration (e.g. Name, Birthdate, Email_id, Phone_No). When user clicks on submit button, data will be inserted into database and user will be redirected to another form. In this form data will be selected from database and displayed in proper format.
4.	**A Data Driven Servlet Application** Develop a data driven servlet application for user authentication. Also display header values of request object and response object.
5.	**Servlet - Session Management** Develop a servlet application for implementing student grading system. In this system on first page student will enter his data (name, id and department). By clicking on next button he will be redirected to another page. On the second page he will enter marks of all 6 subjects. By clicking on "Generate result" he will be redirected to next page. On this page all information of student will be displayed along with the result (pass or fail) in appropriate color. Also display the name of student on each page he visits. (Use session management)
6.	**Servlet - Request Redirection** Develop an application for the following: User should be redirected to the URL entered in location text box.
7.	**JSP - User Authentication** Develop a data driven JSP application for user authentication.
8.	**JSP - Java Beans**

	Develop an application for Bank Account information using Java Beans. This application will perform various operation on bank account like withdraw, deposit and displaying the information.
9.	**JSP - Custom Tags** Develop a custom JSP tag which accepts 10 numbers from user and sorts them in specified order.
10.	**JSP - Application Context** Develop an application that will allow maximum 3 users to access the application. If number of users exceeds than limit then user will be redirected to the error page.
11.	**Simple RESTful Service** Develop a simple JAX-RS service that provides currency conversion.
12.	**Data Driven RESTful Service** Develop a RESTful service to accept student registration data sent from a web form.
13.	**RESTful Web Service - CRUD Operations** Develop a JAX-RS service that provides CRUD database operation interface for a phonebook. Call appropriate service from servlet/jsp pages to perform data oriented operations.
14.	**JAX-RS Client** Develop a JAX-RS client that consumes RESTful service developed in Program-11. Utilize the client in UI layer (JSP pages).
15.	**JAX-RS Client** Develop a RESTful service to perform authentication of the user. The JAX-RS client will be utillized in servlet and provide interface between jsp page and web service to perform authentication.
16.	**Struts2 - User Authentication** Develop a struts 2 application that accepts user name and password from user. If both values are valid then user will be redirected to home page otherwise redirected on error page.
17.	**Struts2 - User Registration and Authentication** Develop a data driven Struts 2 application that accepts registration details from user, i.e. name, contact number, address, user id and password. After successful registration, application redirects user to log in page. If user provides valid authentication details then application will redirect user to home page and will display welcome message, otherwise user will be redirected to error page.
18.	**Struts2 - Data Driven Application** Develop a phone book using struts 2. Application accepts data of contacts like name, address, contact number and email id. The application should provide features to update, delete and list data of contacts into/from database.
19.	**Struts2 - Interceptors**

	Create a logging service using struts 2 to monitor log in and log out timestamp of the users. (Hint: utilize interceptors)
20.	**Struts2 - Hibernate Integration** Develop an application using struts 2 that provides insert, update, delete and list operations for Student data like, name, roll number, semester, course, result (%). The data oriented operations must be performed through hibernate.

Course: MCA **Semester:** 4

Prerequisite: Basic concept of computer organization and knowledge of programming skills.

Rationale : To provide thorough understanding of networking concepts and knowledge of OSI layer's functionality.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
4	0	2	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction** Components of data communication, Reference model - Open System Interconnection (OSI), Transmission Control Protocol/ Internet Protocol (TCP/IP) and comparisons, Types of networks-Local Area Network, Metropolitan Area Network, Wide Area Network, Wireless networks, Internetworks, Data communication fundamentals - introduction, frequency, band, analog signals, digital signals and transmission, Multiplexing and de-multiplexing - Time Division Multiplexing, Frequency Division Multiplexing, Wavelength Division Multiplexing, Orthogonal Frequency Division Multiplexing, Transmission and errors.	15	10
2	**Physical Layer** Transmission media - twisted pair, coaxial cable, fiber optics, Wireless transmission - radio, microwave, infrared, Switching - circuit switching, packet switching, message switching, Analog modulation - Amplitude Modulation, Frequency Modulation, Phase Modulation, Digital modulation - Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying.	15	9
3	**Data Link Layer** Design issues, Error detection and correction, link protocols, Sliding window protocols, Medium access sub layer - channel allocations, ALOHA Protocols, Carrier Sense Multiple Access Protocols (CSMA), CSMA with Collision Detection, Collision free protocols, IEEE 802 standards, Data link layer switching.	20	12
4	**The Network Layer** Design issues, Routing algorithms - shortest path, flooding, Distance Vector Routing, hierarchical, broadcast, multicast, Congestion control algorithms - leaky bucket, token bucket, load shedding, Error detection and correction, Internetworking, IP protocol, Address Resolution Protocol, Reverse Address Resolution Protocol, IP address and classifications.	25	12
5	**Transport Layer** Design issue, Connection management, Transmission Control Protocol (TCP) - introduction, window management, User Datagram Protocol, Performance issues.	15	6
6	**Application Layer** Domain Name Services, Electronic mail, WWW, Hypertext Transfer Protocol, Network security - basics of cryptography and compression techniques.	10	5
	Total	100	54

Reference Books

1.	**Computer Networks (TextBook)** By Andrew. S. Tanenbaum | Prentice Hall Publication, 2004, | 4th edition
2.	**Data and Computer Communication** By W. Stallings | McMillan
3.	**Data Communication and Networking** By Behrouz Forouzan | TMH
4.	**Computer Networks and Internet** By Comer | PHI

Course Outcome

After Learning the Course the students shall be able to:

1. define and explain basics of data communications and computer network architecture.
2. describe OSI reference model and basic functionalities of DNS, WWW and HTTP.
3. identify relevant data transmission techniques and media.
4. implement framing, error handling and congestion control techniques.

List of Practical

1.	**Introduction to LINUX environment and related system programming** Linux provides a complete UNIX programming environment which includes all of the standard libraries, programming tools, compilers, and debuggers which you would expect of other UNIX systems. With Linux, you have access to the complete set of libraries and programming utilities and the complete kernel and library source code. Within the UNIX software world, systems and applications are often programmed in C or C++. The standard C and C++ compiler for Linux is GNU gcc.
2.	**Introduction of various networking equipments and Configuration of Computer Network**
3.	**Introduction to pipes and related system calls for pipe management**
4.	**Framing Protocol: Character Count** This method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where the end of the frame is.
5.	**Framing Protocol: Byte Stufing** Implement a Program in C which demonstrates byte-stuffing framing technique, where sender reads data from a text file, encapsulates the read data in a frame, applies byte stuffing to the frame and sends it to receiver. Assume appropriate character for flag byte and stuff byte. Sender should display the frame data before stuffing and after stuffing for each frame transmitted. Receiver should display the received frame data before de-stuffing and after de-stuffing for each frame. At the end, the data file contents of the sender and the receiver should match. For sender/receiver communication use IPC mechanism FIFO/Named Pipes in Linux/Unix environment. Use Bit-Wise operators in C wherever applicable. Consider a simplistic frame structure consisting of only flag bytes and frame data.
6.	**Framing Protocol: Bit Stuffing** Implement a Program in C which demonstrates bit-stuffing framing technique, where sender reads data from a text file, encapsulates the read data in a frame, applies bit stuffing to the frame and sends it to receiver. Consider the character 01111110 (binary) for flag byte. Sender should display the frame data before bit stuffing and after bit stuffing for each frame

	transmitted. Receiver should display the received frame data before de-stuffing and after de-stuffing for each frame. Receiver should keep on storing the de-stuffed frame data in an external file. At the end, the data file contents of the sender and the receiver files should match. For sender/receiver communication use IPC mechanism FIFO/Named Pipes in Linux/Unix environment. Use Bit-Wise operators in C wherever applicable. Consider a simplistic frame structure consisting of only flag bytes and frame data and neglect all other fields like FCS, etc. Use appropriate data-types for various variables/frame structure. Take appropriate frame size and file sizes. Test the program for arbitrary bit sequences in the sender side data file.
7.	**Error Detection : LRC and Checksum** Implement a Program in GNU C which demonstrates the Block Parity (LRC) Method (for Even Parity). The sender reads an ASCII character string from K/B, applies the block parity logic as applicable, encapsulates the character string in a frame and transmits the frame to the receiver. Sender should display the ASCII character string on screen before and after applying the block parity logic. Receiver should apply the necessary block parity logic and subsequently decide whether there is error or not. If, as per the receiver, there is an error, it should display an appropriate error message on screen else it should display the original character string (as per the receiver). For sender/receiver communication use IPC mechanism FIFO/Named Pipes in Linux/Unix environment. Use Bit-Wise operators in C wherever applicable. Assume a simplistic frame structure consisting of only frame data (ASCII character string). All other fields of frame are to be neglected. Use appropriate data-types for various variables/frame structure. Take appropriate frame size.
8.	**Error Detection : VRC** Implement a Program in GNU C which demonstrates the Single Bit Even Parity Method (VRC). The sender reads a single ASCII character from K/B, applies the parity bit logic as applicable, encapsulates the character in a frame and transmits the frame to the receiver. Sender should display the ASCII character on screen before and after applying the parity bit logic. Receiver should apply the necessary parity logic and subsequently decide whether there is error or not. If, as per the receiver, there is an error, it should display an appropriate error message on screen else it should display the original character (as per the receiver). Create Bit corruption routines on the sender side in the program and accordingly demonstrate those bit error scenarios for which this method works and also those bit error scenarios for which this method fails. All other fields of frame are to be neglected. Use appropriate data-types for various variables/frame structure. Take appropriate frame size.
9.	**Error Detection : CRC** Implement a Program in GNU C which determines whether a given Divisor is valid for CRC. The Divisor is to be entered in binary format from K/B. The sender encapsulates this data in a frame and transmits it to the receiver. The receiver applies the validation logic for CRC divisor and accordingly displays appropriate validation message on screen. For sender/receiver communication use IPC mechanism FIFO/Named Pipes in Linux/Unix environment. Use Bit-Wise operators in C wherever applicable. Consider a simplistic frame structure which encapsulates only the divisor information. All other fields of frame are to be neglected. Use appropriate data-types for various variables/frame structure. Take appropriate frame size.
10.	**Error Correction : Hamming Code** Implement a Program in GNU C which demonstrates the Hamming Code method. The sender reads an ASCII character from K/B, computes the corresponding Hamming Code and encapsulates the Hamming codeword into a frame and transmits the frame to the receiver. Sender should display the bit pattern of the codeword on screen before transmission. Receiver should apply the necessary Hamming code logic and identify whether there is an error or not. In case of no error, it should display the original character on screen. In case of error, it should display appropriate error message and also the bit position of the corrupted bit. Receiver should correct the error and then display the corrected character on screen. Create single Bit corruption routines on the sender side in the program and accordingly test the program for arbitrary single bit error positions and record your observations. For sender/receiver communication use IPC mechanism FIFO/Named Pipes in Linux/Unix environment. Use Bit-Wise operators in C wherever applicable. Consider a simplistic frame structure consisting of only the Hamming Code Word for the given ASCII character
11.	**Congestion control protocols : Leaky Bucket** The leaky bucket is an algorithm used in packet switched computer networks and telecommunications networks. It can be used to check that data transmissions, in the form of packets, conform to defined limits on bandwidth and burstiness (a

	measure of the unevenness or variations in the traffic flow). It can also be used as a scheduling algorithm to determine the timing of transmissions that will comply with the limits set for the bandwidth and burstiness: see network scheduler. The leaky bucket algorithm is also used in leaky bucket counters, e.g. to detect when the average or peak rate of random or stochastic events or stochastic processes exceed defined limits.
12.	**Congestion control protocols : Token Bucket** Unlike leaky bucket, token bucket allows saving, up to maximum size of bucket n. This means that bursts of up to n packets can be sent at once, giving faster response to sudden bursts of input. An important difference between two algorithms: token bucket throws away tokens when the bucket is full but never discards packets while leaky bucket discards packets when the bucket is full. Let token bucket capacity be C (bits), token arrival rate ρ (bps), maximum output rate M (bps), and burst length S (s) – During burst length of S (s), tokens generated are ρS (bits), and output burst contains a maximum of C + ρS (bits) – Also output in a maximum burst of length S (s) is M · S (bits), thus C + ρS = MS or S = C M − ρ Token bucket still allows large bursts, even though the maximum burst length S can be regulated by careful selection of ρ and M. One way to reduce the peak rate is to put a leaky bucket of a larger rate (to avoid discarding packets) after the token bucket.

Miscellaneous

Useful Links

https://sites.google.com/a/paruluniversity.ac.in/pu-dcn/

Course: MCA **Semester:** 4

Prerequisite: Basic knowledge of computer fundamentals, procedural programming, internet concepts, data structures, operating systems, databases, and problem-solving techniques.

Rationale : Student will be assessed on the basis of various courses learned till the end of respective semester.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	-	3	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Fundamentals of Computer Organization** Number systems, Boolean algebra, Flip-Flop, Adders, RAM, ROM, Cache memory, Buses.	10	4
2	**Fundamentals of Programming** Decision & control structure, Looping, Arrays, Functions, Pass by value, Pass by pointer, Pointers, Structure, Union, File handling.	10	5
3	**Fundamentals of Internet and Web Designing** HTML tags, Linking documents, Frames, Media elements, Image maps, CSS styles, Java script.	10	4
4	**Discussion & Q/A**		
5	**Object Oriented Concepts and Programming** Object Oriented Programming concepts, Functions, Constructor and destructor, Operator overloading, Inheritance, Runtime polymorphism, Exception handling, IO operations.	10	4
6	**Operating Systems** Operating System concept, Process Management and threads, Memory Management, Inter-process communication and synchronization, I/O and file system, Disk scheduling.	10	5
7	**Java Programming** Inheritance, Sub classing, Package, Exception, Collection frameworks – ArrayList, Vector, Set, Map, I/O and Multi threading.	10	4
8	**Discussion & Q/A**		
9	**Software Engineering** Software process, Requirements engineering, Data dictionary, Class diagram, Usecase diagram, Activity diagram, Sequence diagram.	10	5
10	**Advanced Database Management Systems** Object relational database, Extended relational database, PL/SQL, Cursor, Trigger, Stored procedure, Database tuning, indexing and security, Backup and recovery in database, Granting and revoking privileges.	10	5
11	**Open Source Technology using PHP** Open source software and configuration, PHP structure and syntax, Session and cookies, MySQL with PHP, working with User data and Error handling.	10	4
12	**Analysis and Design of Algorithm** Algorithm analysis, Time and Space complexity, Divide and conquer algorithms, Graph searching and traversal, Branch and bound, Dynamic programming, Backtracking, Asymptotic notations.	10	5
13	**Discussion & Q/A**		
	Total	100	45

Reference Books

1.	**Structured Computer Organization** By A. Tanenbaum | Prentice - Hall of India Pvt. Ltd | 4th Edition
2.	**Computer Architecture and Logic Design** By Thomas C. Bartee | Tata McGraw - Hill Edition, 2010
3.	**Programming with problem solving through 'C'** By ELSEVIER
4.	**Programming in C** By E. Balagurusamy | Tata McGraw-Hill
5.	**HTML & CSS: The Complete Reference** By Thomas A. Powell | McGraw Hill
6.	**Web Enabled Commercial Application Development using HTML, JavaScript, DHTML and PHP** By Ivan Bayross | BPB Publication
7.	**Database System Concepts** By Silberschatz, Korth, Sudarshan | McGraw Hill Publication | 4th Edition
8.	**Database Systems, Concepts, Design and Applications** By S K Singh | Pearson Edition
9.	**An Introduction to Data Structures with Applications** By Jean-Paul Tremblay, Paul G. Sorenson | Tata McGraw-Hill | 2nd Edition, (2007)
10.	**Introduction to Algorithm** By Cormen, Leiserson, Rivest, Stein | PHI (2003) | 2nd Edition
11.	**C++: The Complete Reference** By Herbert Schildt | McGraw Hill | 5th Edition
12.	**Programming with ANSI C++** By Bhushan Trivedi | Oxford University Press | Second Edition
13.	**Operating System Concepts** By Abraham Siberschatz | Wiley | Fifth
14.	**Operating Systems Internals and Design Principles** By William Stallings | PHI | 5th Edition
15.	**Analysis and Design of Information system** By James Senn | Mc Graw Hill
16.	**System Analysis and Design** By S. Parthasarthy & B. W. Khalkar | Master Ed. Cons., Nashik. | 1st Edition
17.	**The Complete Reference Java J2SE** By Herbert Schildt | TMH Publishing Company Ltd, NewDelhi. | 5th Edition
18.	**Core Java Volume 1** By Cay Horstmann and Gary Cornell | Pearson Education | 8th Edition
19.	**Software Engineering : A Practitioner's Approach** By Pressman R.S | TMH
20.	**Software Engineering** By Sommerville | Pearson | 8th
21.	**SQL, PL/SQL – The Programming Language** By Ivan Bayross | BPB Publications
22.	**Database System Concepts** By Abraham Silberschatz, Henry Korth, S. Sudarshan | McGraw Hill International | 6th Edition
23.	**Professional LAMP: Linux, Apache, MySQL and PHP5 Web Development** By Jason Gerner, Morgan Owens, Elizabeth Naramore, Matt Warden | WROX publication
24.	**Sams Teach Yourself PHP, MySQL and Apache All in One** By Julie C Meloni | Pearson Education
25.	**Analysis and Design of Algorithm** By Ullman | TMH

26.	**Introduction to the Design & Analysis of Algorithms** By Goodman | TMH

Course Outcome

After Learning the Course the students shall be able to:

1. Explain fundamental concepts of computer organization, operating systems, and algorithm analysis including memory, processes, and complexity.
2. Apply procedural and object-oriented programming concepts using C/C++/Java for solving computational problems.
3. Design and develop web-based and database-driven applications using HTML, CSS, JavaScript, PHP, and advanced DBMS features.
4. Analyze and model software systems using software engineering principles, UML diagrams, and efficient algorithms.

Miscellaneous

Useful Links

https://sites.google.com/a/paruluniversity.ac.in/pu_mca_ca/

Parul® University

Subject Syllabus

05201256 - Graphics Programming

Course: MCA (A.Y.-II) 2020 **Semester:** 2

Prerequisite: Knowledge of linear algebra, calculus and programming skills in C/C++.

Rationale : To provide the fundamental principles and algorithms of underlying computer graphics, including line drawing algorithms, circle/ellipse drawing algorithms, 2D geometrical transformation, 3D geometric transformations, viewing in 3D (orthographic projection and perspective projection) and visible surface detection algorithms.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	-	2	-	4	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction and Basic Drawing Algorithms** Computer graphics and its applications, Raster-scan system, Randomscan system, Graphics primitives. Basic Drawing Algorithms: Line-drawing algorithms - DDA Algorithm, Bresenham's line algorithm. Circle Generating Algorithms: Midpoint circle algorithm, Bresenhams circle algorithm. Ellipse-Generating Algorithm: Midpoint ellipse algorithm.	**20**	**8**
2	**Region Filling Algorithms and Two-Dimensional Geometric Transformations** Attributes of output primitives, Line style, Colour and intensity, Area filling algorithm, Scan line and boundary fill algorithm, Flood fill algorithm, Antialiasing technique. Two dimensional transformations: Translation, Scaling, Rotation, Reflection, Sheering, Composite transformation, Transformation commands, Character generation.	**20**	**12**
3	**Viewing and Clipping**Viewing coordinates, Window view port, Clipping, Window to view port transformation, Line clipping algorithm - Cohen Sutherland, Polygon clipping - Sutherland Hodgman algorithm, 3D clipping - normalized view volume, View port clipping, Clipping in homogeneous coordinates.	**20**	**7**
4	**Illumination Model** **Light sources, Diffuse reflection, Colour Models like RGB, YIQ, CMY, HSV.** **3D Viewing: 3D concepts, 3D display techniques, 3D representation, 3D transformation - transition, scaling, composite transformation, rotation about arbitrary axis.** **Projection: Parallel and perspective projection, Hidden surface and line removal, Back face removal, Depth buffer and scan line methods**	**20**	**7**
5	**OpenGL and Graphics Programming**Introduction to OpenGL: Basics of OpenGL, its pipeline, and setup, OpenGL coordinate system. OpenGL Programming Basics: Drawing shapes, event handling, and rendering pipeline Lighting and material properties in OpenGL. Texture Mapping & Shading: Texture mapping techniques, OpenGL texture functions, shading techniques (Flat, Gouraud, and Phong), basics of OpenGL shaders (GLSL). OpenGL Animations & Rendering: Animation concepts, double buffering, frame rate control, shadow mapping. Advanced OpenGL & Project: Introduction to modern OpenGL (VBOs, VAOs, and shaders), optimization techniques	**20**	**14**
	Total	**100**	**48**

Reference Books

1.	**Computer Graphics** By Donald Hearn & M. Pauline Baker | PHI,2011 | Second Edition
2.	**Computer Graphics - a Programming approach** By S. Harrington | McGraw Hill,2014 | 2nd Edition
3.	**Principles of interactive computer graphics** By New Mann & Sprovl | McGraw Hill,2001 | 2nd edition
4.	**Procedural Elements for Computer Graphics** By David F. Rogers | Tata McGraw Hill,2001
5.	**Multimedia System Design** By Prabhat K. Andleigh & Kiran Thakur | PHI
6.	**Computer Graphics with Open GL** By Hearn Baker Carithers | Person New International Edition | 4th
7.	**Learn Open GL Graphics Programming** By Joey de Vries

Course Outcome

After Learning the Course the students shall be able to:
1. Identify various components of the graphics system. 2. Implement algorithms for rendering basic shapes. 3. Describe the significance of projection, illumination models, viewing and clipping techniques. 4. Implement various 2D and 3D transformation techniques and clipping algorithms.

List of Practical

1.	**Implement and visualize a line using the Digital Differential Analyzer (DDA) algorithm.**
2.	**Implement the Bresenham's line drawing algorithm to render an efficient line.**
3.	**Draw a circle using the Midpoint Circle Algorithm.**
4.	**Implement Bresenham's circle drawing algorithm to generate a circle.**
5.	**Draw an ellipse using the Midpoint Ellipse Algorithm.**
6.	**Implement Scan-line and Boundary Fill algorithms for polygon filling.**
7.	**Implement Flood Fill Algorithm to fill a closed shape with color.**
8.	**Perform translation, scaling, rotation, reflection, and shearing on 2D objects.**
9.	**Implement the Cohen-Sutherland algorithm for line clipping within a viewport.**
10.	**Implement the Sutherland-Hodgman algorithm for polygon clipping.**
11.	**Perform 3D translation, scaling, rotation, reflection, and shearing.**
12.	**Implement Back-face culling, Depth-buffer (Z-buffer), and Scan-line algorithms for hidden surface removal.**
13.	**Implement RGB, HSV color models and anti-aliasing techniques.**
14.	**Apply shading models and textures to 3D objects.**
15.	**Create an animated scene using transformation functions.**

Course: MBA **Semester:** 2

Prerequisite: Knowledge of Professional Skills - I

Rationale : This course develops advanced communication and employability skills for effective professional performance and career readiness.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			Marks EYE		
					T	CE	P	T	P	
1	-	2	-	2	40	-	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Picture Perception**Concept of Picture Perception and Interpretation Observation, Assumptions, and Logical Thinking Story Structure: Past–Present–Future	5	1
2	**LinkedIn Profiling and Resume Building – Difference between CV/Resume; Format, Heading and Application Procedure:**Difference between CV and Resume Resume Structure, Headings, and Content LinkedIn Profile Essentials, Key words and Professional Branding	15	2
3	**Writing Skills: Report, Memo & Proposal Writing**Purpose and Types of Reports, Memos, and Proposals Structure, Format, and Language Style Clarity, Objectivity, and Professional Tone	10	1
4	**Decision-Making & Problem Solving through Case Studies**Decision-Making Process and Models Problem Identification and Root Cause Analysis Ethical and Logical Decision-Making	5	1
5	**Reading Comprehension (Level of Difficulty - Advanced & Case Study based)**Advanced Reading Strategies (Inference, Evaluation) Understanding Case Study based Passages Critical Reading and Logical Reasoning	10	1
6	**Organizing Meetings - Agenda, Minutes of Meeting**Purpose and Types of Meetings Agenda Preparation and Meeting Roles Format and Language of Minutes of Meeting	10	1
7	**JAM Sessions**Concept and Purpose of JAM Content Structuring under Time Limit Fluency, Coherence, and Confidence	5	1
8	**Time Management and Stress Management**Importance of Time Management Prioritization Techniques (Urgent vs Important) Pomodoro technique and Eisenhower Matrix	10	1
9	**Group Discussions: Do's and Don'ts - Practice Sessions:**Purpose and Types of Group Discussions Roles, Etiquette, and Evaluation Criteria	10	2

	Common Mistakes in GDs		
10	**Interview Skills – STAR Technique- Mock interview sessions:**Types of Interviews STAR Technique for answering behavioural questions	**20**	**4**
	Total	**100**	**15**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate
Weightage	15	15	35	15	20

Reference Books

1.	**The 7 Habits of Highly Effective People** By Stephen R. Covey | Simon & Schuster, Pub. Year 2020
2.	**Basic Business Communication** By Lesikar, Petit &Flately, Lesikar's | Tata McGraw Hill
3.	**Basic Communication** By Poe & Fruchling | AITBS
4.	**Personality Development and Soft Skills** By Barun K Mitra | Oxford, 2011
5.	**1. Sanjay Kumar, Pushp Lata, Communication Skills, Oxford University Press**

Course Outcomes

At the end of this course Students Will be able to:	
1	Explain key formats, structures, and terminology related to professional documents and workplace communication.
2	Apply communication strategies, case analysis, and the STAR technique in group discussions, interviews, and professional situations.
3	Analyse professional documents, case studies, and workplace interactions to identify effectiveness and areas for improvement.
4	Assess personal and peer performance in mock interviews and documentation tasks, and group discussions using defined criteria.
5	Design and develop structured reports, memos, proposals, resumes, CVs, LinkedIn profiles, meeting agendas, and minutes of meetings demonstrating professional standards.

List of Practical	
1.	**Picture Perception** Picture-based Story Writing Group Discussion on Picture Interpretation
2.	**Writing Skills: Report, Memo & Proposal Writing** Writing a Short Report on a Given Case Drafting Internal Memos for Workplace Scenarios Preparing a Proposal
3.	**Decision-Making & Problem Solving through Case Studies** Case Study Analysis in Groups Problem–Solution Presentation Decision Justification Exercise
4.	**Reading Comprehension – Advanced & Case Based** Advanced RC Practice Sets Case study based Group Discussion on Case Conclusions
5.	**LinkedIn and Resume Building – Difference between CV/Resume; Format, Heading and Application Procedure** Resume Drafting Activity LinkedIn Profile Creation/Optimization
6.	**Organising Meetings - Agenda, Minutes of Meeting** Agenda Drafting Exercise Mock Meeting Role-play Writing Minutes of Meeting
7.	**JAM Sessions** Individual JAM Speaking Rounds Topic-based JAM Practice
8.	**Time Management and Stress Management** Time Log and Priority Planning Activity Stress Management Techniques Practice Group Discussion on Personal Challenges
9.	**Group Discussions: Do's and Don'ts - Practice Sessions** Mock Group Discussion Sessions Observation and Feedback Activity
10.	**Interview Skills – STAR Technique- Mock interview sessions** STAR-based Answer Framing Mock Interview Sessions Feedback and Improvement Discussion

Course: MCA (A.Y.-II) 2020

Semester: 2

Prerequisite: Knowledge of data structures and discrete mathematics.

Proposed Rationale: To provide fundamental knowledge of Artificial Intelligence including problem solving, knowledge representation, and reasoning techniques. To introduce data handling and basic machine learning concepts for real-world applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	1	2	-	5	40	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Sr.	Topics	W	T
1	**Introduction To AI & Real-World Applications:** History & evolution of AI, Types: ANI, AGI, ASI, AI applications (healthcare, education, business), Turing Test, Introduction to modern AI (Generative AI overview)	**10**	**6**
2	**Intelligent Agents & Environments:** Agents & environment, Types of agents: Simple reflex, Goal-based, Utility-based Learning agents, Rationality & performance measures How AI Agent works	**10**	**6**
3	**Problem Solving & Search Techniques:** Problem formulation & state space, Production systems Informed search (Heuristics, A*, Hill Climbing, Generate and Test, Means Ends Analysis, Constraints Satisfaction Problem) Game playing: Minimax, Alpha-Beta pruning	**20**	**9**
4	**Knowledge Representation & Logic** Knowledge representation: Introduction, Types of Knowledge, Approaches Propositional logic Predicate logic Inference: Forward chaining, Backward chaining	**15**	**6**
5	**Reasoning Under Uncertainty** Probability basics, Bayes theorem, Bayesian networks, Fuzzy logic	**15**	**5**
6	**Data Handling & Feature Engineering** Types of data (structured/unstructured), Data preprocessing (cleaning, normalization), Feature selection & extraction, Handling imbalanced data	**15**	**6**
7	**Introduction To Machine Learning** Learning from data Types: Supervised, Unsupervised, Reinforcement learning Models: Linear regression, Logistic regression, SVM	**15**	**7**
	TOTAL	**100**	**45**

Course Outcomes
After Learning the Course, the students shall be able to:
1. Explain fundamental concepts of Artificial Intelligence, including its types and applications. 2. Analyse intelligent agents and their interaction with different environments. 3. Apply problem-solving and search techniques to AI-based problems. 4. Demonstrate knowledge representation and logical reasoning using appropriate methods. 5. Evaluate uncertainty models and apply basic machine learning techniques with data handling.

List of Practical	
1.	Implement a simple reflex agent for a vacuum cleaner environment using condition–action rules.
2.	Modify the above program to implement a goal-based agent.
3.	Implement a Tic-Tac-Toe game using AI concepts.
4.	Develop a rule-based system for simple medical diagnosis.
5.	Write a Python program to solve the Water Jug Problem.
6.	Perform data cleaning and preprocessing on datasets by handling missing values, normalization, and feature transformation.
7.	Perform Linear Regression for modeling and predicting continuous variables. *Sample Project:* House Price Prediction using Linear Regression OR ANY
8.	Perform binary classification using Logistic Regression on structured data. *Sample Project:* Email Spam Detection OR ANY
9.	Perform classification using Support Vector Machine (SVM) on a dataset and evaluate its performance. *Sample Project:* Breast Cancer Detection using SVM OR ANY
10.	Case Study: Analyze a real-world AI application (e.g., healthcare diagnosis system) and explain how agents, search, or ML techniques are used.

Course: MCA

Semester: 2

Prerequisite: Fundamental knowledge of computer network.

Rationale : The key objectives of this course are to develop an understanding of information assurance as practiced in computer operating systems, networks and representative applications and to gain familiarity with prevalent attacks, defenses against them.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	1	2	-	5	40	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content — **W** - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Cyber Security Foundations and Threat Landscape** Introduction to Cyber Security, Information Security principles, CIA Triad, Security goals and models, Defense-in-depth strategy, Types of cyber threats and attacks (Malware, Phishing, Ransomware, Social Engineering), Risk management concepts, Security policies, Security standards and frameworks (NIST, ISO 27001, CIS Controls). **Cryptography and Secure Communication** Fundamentals of cryptography, Symmetric encryption (AES, DES), Asymmetric encryption (RSA), Cryptographic hash functions (SHA-256, SHA-3), Digital signatures, Message authentication codes (MAC), Public Key Infrastructure (PKI), Introduction to AI Security, post-quantum security.	24	12
2	**Network Security and Monitoring** Network security architecture, Firewalls and access control lists, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Virtual Private Networks (VPN), Network security protocols, Network traffic monitoring and packet analysis, Introduction to Security Information and Event Management (SIEM). **System and Application Security** Operating system security mechanisms, Linux and Windows security architecture, Privilege management and access control, Vulnerability assessment and patch management, Secure configuration and hardening.	20	10
3	**Cloud Security** Introduction to cloud computing and cloud security, Cloud service models (IaaS, PaaS, SaaS), Cloud deployment models (Public, Private, Hybrid), Shared responsibility model, Identity and Access Management (IAM), Multi-Factor Authentication (MFA), Cloud security threats (data breaches, misconfigured storage, insecure APIs, account hijacking, insider threats, DDoS attacks), Data protection techniques including encryption and key management, Backup and disaster recovery, Cloud security monitoring and logging, Introduction to cloud security tools such as AWS Security Hub, AWS CloudTrail, AWS GuardDuty, Azure Security Center, and Google Cloud Security Command Center. **Mobile Security** Introduction to mobile security concepts, Mobile operating system architecture, Android security model, Mobile application vulnerabilities and threats, Malware and spyware in mobile devices, Mobile authentication mechanisms, Secure mobile communication, Mobile device management (MDM), Mobile application security	32	16

	testing and analysis. **Virtualization and Container Security** Introduction to virtualization concepts, Types of virtualization, Virtual machines and hypervisors, Security issues in virtualized environments, Containerization concepts and container security, Security challenges in Docker and Kubernetes environments, Best practices for securing virtualized infrastructures.		
4	**Digital Forensics and Incident Response** Introduction to digital forensics, Types of digital evidence, Rules of evidence and chain of custody, Digital forensic investigation process, Disk forensics, Memory forensics, Network forensics, Log analysis, Incident detection and response process, Documentation and reporting of forensic findings.	**12**	**5**
5	**Advanced Digital Forensics and Investigation Techniques** Forensic acquisition and imaging techniques, File system forensics, Email forensics, Malware forensics, Mobile device forensics, Cloud forensics concepts, Timeline analysis, Evidence preservation and integrity verification, Use of digital forensic tools such as Autopsy, FTK Imager, Volatility Framework, and Wireshark for forensic investigation.	**12**	**5**
	Total	**100**	**48**

Reference Books

1.	**Information systems security (TextBook)** By Nina Godbole | Wiley Publications,2008
2.	**Cyber Security understanding Cyber Crimes, Computer forensics and Legal Perspectives (TextBook)** By Nina Godbole and Sunit Belapure
3.	**Cryptography and Network Security Principles and Practices** By W. Stallings | Prentice-Hall of India, 2006 | 4th Edition
4.	**Information Security: Principles and Practices** By M. Merkow and J. Breithaupt | Pearson Education, 2006

Course Outcome

After Learning the Course the students shall be able to:

1. **Understand and analyze cybersecurity principles, threat landscapes, and security frameworks** used to protect modern information systems.
2. **Apply cryptographic techniques and network security mechanisms** to secure communication, detect attacks, and protect network infrastructure.
3. **Evaluate and implement security practices for systems, cloud platforms, mobile devices, and virtualized environments** to mitigate vulnerabilities and cyber threats.
4. **Conduct digital forensic investigations and incident response activities** by analyzing digital evidence using forensic tools and methodologies.

List of Practical

1.	**Cyber Security Lab Setup** **Installation and configuration of virtual lab environment using VirtualBox / VMware, installation of Kali Linux, setup of vulnerable machines such as Metasploitable and OWASP BWA, and understanding the role of Kali Linux in cybersecurity and forensic analysis.**
2.	**Network Scanning and Enumeration** **Perform host discovery, port scanning, and service enumeration using Nmap, Netdiscover, and Recon-ng.**
3.	**Packet Capture and Network Traffic Analysis** **Capture and analyze network packets to understand network communication and detect suspicious traffic using Wireshark and Tcpdump.**
4.	**Vulnerability Assessment and Scanning** **Identify system and web vulnerabilities using OpenVAS, Nessus, and Nikto.**

5.	**Web Application Security Testing** **Perform web application penetration testing to detect vulnerabilities such as SQL Injection and Cross-Site Scripting using Burp Suite, OWASP ZAP, and DVWA (Damn Vulnerable Web Application).**
6.	**Password Cracking and Authentication Testing with Mitigation** **Perform password attacks such as dictionary and brute-force using John the Ripper, Hydra, and Hashcat, and analyze prevention techniques such as strong password policies and hashing.**
7.	**Penetration Testing using Metasploit Framework** **Exploit vulnerabilities in a controlled lab environment using Metasploit Framework and analyze post-exploitation techniques.**
8.	**Wireless Network Security Testing** **Analyze wireless network security and perform wireless scanning using Aircrack-ng, Airodump-ng, and Kismet.**
9.	**Cloud Security Monitoring using Free AWS Tools** **Configure and analyze cloud security features using AWS IAM, AWS CloudTrail, and AWS GuardDuty (free tier/trial), including access control and logging.**
10.	**Mobile Application Security Analysis** **Analyze Android applications and identify vulnerabilities using MobSF (Mobile Security Framework), APKTool, JADX, and ADB.**
11.	**Malware Analysis Basics** **Perform static analysis of suspicious files using PEStudio, Strings, and VirusTotal.**
12.	**Digital Forensics Investigation** **Acquire and analyze disk images for forensic investigation using Autopsy, FTK Imager, and The Sleuth Kit.**
13.	**Memory Forensics Analysis** **Analyze memory dumps and detect malicious processes using Volatility Framework.**
14.	**Security Log Analysis and Incident Investigation** **Analyze system and network logs using tools like Splunk or ELK Stack to detect suspicious activities and support incident investigation. (Elasticsearch, Logstash, Kibana).**

Course: MCA **Semester:** 4

Prerequisite: Basic knowledge of computer science fundamentals, familiarity with current computing technologies, ability to read and understand research papers/technical articles, and basic presentation and documentation skills.

Rationale : This course/seminar provides students with an opportunity to explore contemporary topics in Computer Science and Applications, enhance their ability to study and analyze technical literature, and develop skills in research, documentation, and professional presentation, thereby preparing them for advanced studies and industry-oriented problem solving.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	-	1	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Seminar** Each Student is expected to choose a topic of current relevance of Computer Science/Application field. They have to refer papers / reading material from various resources including latest tools. Student should compile the material in the form of presentation / seminar report and appropriately present the topic**.**	**100**	
	Total	**100**	

Course Outcome

After Learning the Course the students shall be able to:

1. learn about exploring topic on ICT domain.
2. develop skills for identification and retrieval of relevant information.
3. effectively communicate and present ICT topic.
4. summarize on ICT topic.

Course: MCA (A.Y.-II) 2020 **Semester:** 2

Prerequisite: basic knowledge of HTML, CSS, and JavaScript fundamentals along with familiarity in working with any programming environment or web development tools.

Rationale : This course equips students with advanced JavaScript, modern UI frameworks, ReactJS, API integration, and MongoDB skills needed to build full-stack, industry-ready web applications. It enables learners to develop efficient, responsive, and scalable solutions aligned with current trends in modern web development.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	1	2	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Advanced JavaScript Concepts** Deep Dive into JavaScript Fundamentals (Scope, Hoisting, and Closures, Prototypes and Prototypal Inheritance this Keyword and its Context Execution Context and Call Stack) Asynchronous JavaScript (Promises and Async/Await) JavaScript Design Patterns (Singleton, Factory, Observer, and Strategy Patterns Module Pattern and Revealing Module Pattern Functional Programming) Memory Management Functional Programming	15	7
2	**JavaScript Objects & Arrays** ES6 Features (let/const, Arrow Functions, Template Literals, Destructuring, Spread & Rest Operators), Promises & Async/Await, Fetch API, JavaScript Modules (Import/Export), Error Handling & Debugging, Regular Expressions, Higher-Order Functions (map, filter, reduce), Event Loop & Callbacks, Local Storage & Session Storage, Object-Oriented JavaScript (Classes, Prototypes, Inheritance)	15	7
3	**Advanced UI Design** Material UI Tailwind CSS Bootstrap concepts	15	10
4	**ReactJS**Introduction to React, Setting Up a React Environment, React Components, JSX Syntax, Props and State, Handling Events, Conditional Rendering, Lists and Keys, Forms and Controlled Components, Advanced Component Patterns, React Hooks, Performance Optimization, Context API and State Management, Server-Side rendering and Static site generation, Testing and Best Practice	15	7
5	**API Calling & Integration** RESTful API Basics Fetch API and Axios Handling HTTP Methods (GET, POST, PUT, DELETE) Authentication and Authorization (JWT, OAuth) Error Handling in API Calls	15	7
6	**Database Connectivity with MongoDB** Introduction to NoSQL and MongoDB CRUD Operations in MongoDB Mongoose ODM and Schema Design Aggregation Framework	20	10

	Connecting Node.js with MongoDB		
	Total	**95**	**48**

Reference Books

1.	**You Don't Know JS" (Series)** By Kyle Simpson
2.	**JavaScript: The Good Parts** By Douglas Crockford
3.	**Eloquent JavaScript** By Marijn Haverbeke
4.	**Tailwind CSS: From Zero to Production** By Simon Vrachliotis
5.	**The Road to React** By Robin Wieruch
6.	**Fullstack React** By Anthony Accomazzo
7.	**REST API Design Handbook** By Mike Stowe
8.	**API Design Patterns** By JJ Geewax

Course Outcome

After Learning the Course the students shall be able to:
1: Understand advanced JavaScript concepts including ES6 features, asynchronous programming, and functional programming. 2: Apply modern UI frameworks like Material UI, Tailwind CSS, and Bootstrap to design responsive interfaces. 3: Develop dynamic web applications using ReactJS components, hooks, state management, and best practices. 4: Integrate RESTful APIs and connect applications with MongoDB using CRUD operations and Mongoose.

List of Practical

1.	**Create a function with nested functions demonstrating closures, lexical scope, and use of this in different contexts.**
2.	**Simulate API calls using Promises and rewrite using async/await with proper error handling.**
3.	**Use let, const, arrow functions, destructuring, spread/rest, and template literals in a mini calculation or info display program.**
4.	**Create a task list using map, filter, reduce, and store/retrieve it from local storage.**
5.	**Build a responsive profile or landing page using Tailwind CSS utility classes.**
6.	**Design a product grid using Bootstrap cards with responsive breakpoints.**
7.	**Build a React app using Context API for theme switching and create a custom hook for toggling dark/light mode.**
8.	**Fetch user data from a public API using Axios and display it with proper loading and error messages.**
9.	**Create a simple Express server connected to MongoDB for adding, viewing, and deleting students.**
10.	**Define a Mongoose schema for products and perform an aggregation to group or filter products based on category or price**

Course: MCA (A.Y.-II) 2020 **Semester:** 2

Prerequisite: Fundamental knowledge of Networking and Computing, Basic Knowledge of Web Services

Rationale : The key objectives of this course are to provide an understanding of the basic concepts of parallel and distributed computing and their role in cloud computing, to study the concept of virtualization and relevant technologies available in the market, to understand the importance of cloud computing for higher throughput, to make aware about availability of various cloud platforms, to study different application of cloud and cloud management techniques.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/	Credit	T	CE	P	T	P	
3	1	2	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction** Cloud computing introduction, history, characteristics, Pros and cons of cloud computing, Nature of cloud, Technologies in cloud computing, Migrating into the cloud, Cloud applications Working of cloud computing, Trends in cloud computing.	15	8
2	**Cloud Computing Models** Types of Cloud, Cloud service models, Cloud deployment models, Cloud Service models - Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Other cloud services, Cloud architecture, Cloud computing reference model.	15	8
3	**Virtualization Concepts** Virtualization - definition, architecture, software, applications, Virtual clustering, Anatomy of cloud infrastructure, Virtual infrastructures, CPU virtualization, Network and storage virtualization.	15	8
4	**Cloud Storage** Data Storage - Introduction to cloud data storage, Storage options in Cloud, Structured and Unstructured Storage in the Cloud, Exploring and Cloud Storage Services, data storage management, Cloud data stores, Provisioning cloud storage, Data-intensive technologies for cloud computing, Cloud storage from LANs to WANs - cloud characteristic, distributed data storage, applications utilizing cloud storage.	15	8
5	**Cloud Risks and Cloud Security** Risks in cloud computing - introduction, risk management, cloud impact, enterprise wide risk management, types of risks in cloud. Data security in cloud - Introduction to Cloud Security, digital persona and data security, content level security, Understanding the shared security model Cloud security services, Understand authentication and security authorization, challenges in the cloud, Secure cloud - software requirements, software testing.	20	8
6	**Google Cloud Platform (GCP)** The GCP console, Understanding GCP Projects, Install and Configure Cloud SDK, Use Cloud Shell, GCP APIs, Cloud Console Mobile App, Use GCP to build your Apps: Compute Options in the Cloud Exploring IaaS with Compute Engine [With Lab], Configuring Elastic Apps with Autoscaling.	20	8
	Total	100	48

Reference Books

1.	**Cloud Computing - A practical approach for learning and implementation** By A.Srinivasan and J.Suresh | Pearson Publications
2.	**Cloud Computing: A practical approach** By Anthony T. Vetle | TMH
3.	**Cloud Computing For Dummies** By Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper | Wiley India Pvt Ltd
4.	**Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More** By Kris Jamsa | Jones & Bartlett Learning
5.	**Cloud Computing Bible** By Barrie Sosinsky | Wiley India Pvt Ltd

Course Outcome

After Learning the Course the students shall be able to:

- define cloud computing and its applications.
- describe role of virtualization in establishing cloud.
- understand principles and service models of cloud computing.
- familiarize with various cloud computing platforms.
- conceptualize cloud storage management and data security.
- identify feasible cloud platforms for solutions to be provided.
- deploy services and resources on cloud platforms.

List of Practical

1.	**Cloud Storage: Sign Up for a free account on two or more providers of your choice. Store any 10 files that are larger than 10 MB each in at least two providers of your choice.**
2.	**Cloud Models: Create or Sign Up for PaaS, Saas and IaaS services on platform of your choice and upload at least one of your resource or application on each platform.**
3.	**Para-Virtualization: Implement one VMs in LINUX/Windows using VM software on your local machine.**
4.	**Start a Windows Virtual Machine on Google Compute Engine.**
5.	**Start a LINUX Virtual Machine on Google Compute Engine.**
6.	**Build a Docker Application on GCP. You can use Git Repository to build.**
7.	**Develop, Build and Deploy a container Application On Google Compute Engine.**
8.	**Explore GCP. Perform at least one Case Study on GCP services and prepare detailed report on your Case Study.**

Course: MCA (A.Y.-II) 2020 **Semester:** 2

Prerequisite: Basic Knowledge of Data Analytics and Computation Methods.

Rationale : The objective of this course is to provide Conceptual insight about Big Data Analytics, Installation and understanding of Hadoop Architecture and its ecosystems.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	1	2	-	5	20	20	20	60	30	150

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Overview of Big Data** Introduction to Big Data, Evolution of Big Data, Structuring of Big Data, Fundamentals of Big Data, Big Data Analytics, Career and Future in Big Data	10	5
2	**Discovering the Use of Big Data in Business Context** Big Data in Social Networking, Big Data in Preventing Fraudulent Activities, Big Data in Detecting Fraudulent Activities in Insurance Sector, Use of Big Data in Retail Industry.	10	5
3	**Technologies for Handling Big Data** Distributed and Parallel Computing for Big Data, Introducing Hadoop, Cloud Computing and Big Data, In-Memory Computing Technology for Big Data.	8	4
4	**Understanding Hadoop Ecosystem** History of Hadoop, Hadoop Ecosystem, Analysing Data with Unix tools, Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System, IBM Big Data Strategy, Introduction to Infosphere BigInsights and Big Sheets.	12	6
5	**HDFS(Hadoop Distributed File System)** The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures.	14	6
6	**MapReduce Fundamentals** Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.	14	6
7	**Understanding Big Data Technology Foundations, Storing Data in Databases and Data Warehouses** Understanding Big Data Technology Foundations: Exploring the Big Data Stack, Virtualization and Big Data, Virtualization Approaches, Summary, Quick Revise Storing Data in Databases and Data Warehouses: RDBMS and Big Data, Non-Relational Database, Polyglot Persistence, Integrating Big Data with Traditional Data Warehouses, Big Data Analysis and Data Warehouse, Changing Deployment Models in Big Data Era.	16	8
8	**Processing Your Data with MapReduce** Recollecting the Concept of MapReduce Framework,Developing Simple MapReduce Application, Points to	16	8

	Consider while Designing MapReduce.		
	Total	**100**	**48**

Reference Books

1.	**Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses** By Michael Minelli, Michelle Chambers, and Ambiga Dhiraj | Wiley, Pub. Year 2013
2.	**Big-Data Black Book** By DT Editorial Services | Wiley India
3.	**Hadoop Operations, Eric Sammer, O'Reilley.** By O'Reilley
4.	**Hadoop: The Definitive Guide by Tom White, Third Edition, O'Reilley.** By Tom White

Course Outcome

After Learning the Course the students shall be able to:

1. Work with big data platform and explore the big data analytics techniques business applications.
2. Understand the fundamentals of various big data analytics techniques.
3. Analyse the HADOOP and Map Reduce technologies associated with big data analytics.
4. Have knowledge on accessing, storing and manipulating the huge data from different resources.

List of Practical

1.	**Practical-1** (i)Perform setting up and Installing Hadoop in its two operating modes: • Pseudo distributed, • Fully distributed. (ii) Use web based tools to monitor your Hadoop setup.
2.	**Practical-2** (i) Implement the following file management tasks in Hadoop: • Adding files and directories • Retrieving files • Deleting files ii) Benchmark and stress test an Apache Hadoop cluster
3.	**Practical-3** Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm. • Find the number of occurrence of each word appearing in the input file(s) • Performing a Map Reduce Job for word search count (look for specific keywords in a file)
4.	**Practical-4** To understand the overall programming architecture using Map Reduce API
5.	**Practical-5** Stop word elimination problem: • Input: A large textual file containing one sentence per line A small file containing a set of stop words (One stop word per line)

	• Output: A textual file containing the same sentences of the large input file without the words appearing in the small file.
6.	**Practical-6** Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented. Data available at: https://github.com/tomwhite/hadoopbook/tree/master/input/ncdc/all. • Find average, max and min temperature for each year in NCDC data set? • Filter the readings of a set based on value of the measurement, Output the line of input files associated with a temperature value greater than 30.0 and store it in a separate file.
7.	**Practical-7** Store the basic information about students such as roll no, name, date of birth , and address of student using various collection types such as List, Set and Map
8.	**Practical-8** Basic CRUD operations in MongoDB
9.	**Practical-9** Retrieve various types of documents from students collection
10.	**practical-10** To find documents from Students collection
11.	**Practical-11** Develop Map Reduce Work Application
12.	**Practical-12** Creating the HDFS tables and loading them in Hive and learn joining of tables in Hive