



Andhra Pradesh State Council of Higher Education

Information Technology :Minor

w.e.f AY 2023-24 onwards

COURSE STRUCTURE

Year	Semester	Course	Title	No. Hrs./ Week	No. of Credits	
	II	1	Problem Solving in C - (T)	3	3	
			Problem Solving in C - (P)	2	1	
II	III	2	Database Management Systems- (T)	3	3	
			Database Management Systems- (P)	2	1	
	IV	3	Web Technologies- (T)	3	3	
			Web Technologies- (P)	2	1	
		4	Object Oriented Programming using Java- (T)	3	3	
			Object Oriented Programming using Java- (P)	2	1	
	III	V	5	R Programming- (T)	3	3
				R Programming- (P)	2	1
6		Software Engineering - (T)	3	3		
		Software Engineering - (P)	2	1		

II Semester
Course1: Problem Solving in C

Credits -3

I. LEARNING OUTCOMES: Upon successful completion of the course, a student will be able to:

1. Understand the functionality of a Digital Computer and fundamental constructs of programming.
2. Analyze and develop solutions to a given problem using control statements.
3. Work with arrays and textual information.
4. Understand the concept of functional hierarchical code organization.
5. Gain knowledge on derived data types and file handling.

UNIT I

Introduction to Computer and Programming: Introduction - Block diagram of a computer - Hardware and Software - Generations of Programming Languages - Algorithms - Flowcharts. Introduction to C: Introduction - Structure of C Program - Writing the first C Program - File used in C Program - Compiling and Executing C Programs - Using Comments - Keywords - Identifiers - Basic Data Types in C - Variables - Constants - I/O Statements in C - Operators in C.

UNIT II

Decision Control and Looping Statements: Decision making statements: if, else if, else if ladder, switch statements; Loop Control Statements: while, do-while, for loop; break, continue and goto statements.

UNIT III

Arrays: Introduction - One Dimensional Arrays - Declaration, Initialization and Memory representation; Two Dimensional Arrays - Declaration, Initialization and Memory Representation; Strings: Declaring and Initializing string variables, character and string handling functions.

UNIT IV

Functions: Introduction - Function declaration/ prototype - Function definition - function call - return statement - Categories of functions - Recursion - Parameter Passing techniques - Scope of variables - Storage Classes.

Pointers: Introduction to Pointers - declaring and initializing pointer Variables - accessing values using pointers - Pointer Arithmetic - Dynamic Memory Allocation.

UNIT V

Structures and Unions: Introduction - Structure definition - accessing structure members - Array of Structures - union definition - difference between structures and unions.

Files: Introduction to Files - Using Files in C - Reading Data from Files - Writing Data to Files - Detecting the End-of-file - Accepting Command Line Arguments.

III. REFERENCES:

TEXT BOOKS:

1. E Balagurusamy - Programming in ANSI C - Tata McGraw-Hill publications.
2. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

REFERENCE BOOKS:

1. Brain W Kernighan and Dennis M Ritchie - The 'C' Programming language - Pearson Publications.
2. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publications.
3. Yashavant Kanetkar - Let Us 'C' - BPB Publications.

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside

the syllabus content. Shall be individual and challenging)

2. Student seminars (on topics of the syllabus and related aspects (individual activity))

a. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))

b. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

II Semester
Course 1: Problem Solving in C
Credits -1

IV. PROBLEM SOLVING IN C – PRACTICAL

1. Write a program to find the area of circle and triangle.
2. Write a program to find simple and compound interest.
3. Write a program to convert temperature from Celsius to Fahrenheit
4. Write a program to find whether a number is even or odd
5. Write a program to find sum and average of 5 numbers
6. Write a program to check whether the given number is Armstrong or not.
7. Write a program to find the sum of individual digits of a positive integer.
8. Write a program to generate the first n terms of the Fibonacci sequence.
9. Write a program to find both the largest and smallest number in a list of integer values
10. Write a program to calculate factorial of given integer value using recursive functions
11. Write a program for addition of two matrices.
12. Write a program for multiplication of two matrices.
13. Write a program to perform various string operations.
14. Write a program to search an element in a given list of values.
15. Write a C program to write and read data into/from a File.

III Semester
Course 2: Database Management Systems
Credits -3

I. LEARNING OUTCOMES:

On successful completion of the course, students will be able to

1. Differentiate between database systems and file based systems
2. Design a database using ER model
3. Use relational model in database design
4. Use SQL commands for creating and manipulating data stored in databases.
5. Write PL/SQL programs to work with databases.

II. SYLLABUS:

UNIT I

Overview of Database Management System: Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of data base.

UNIT II

Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, advantages of ER modeling.

UNIT III

Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, Functional dependencies and normal forms upto 3rd normal form and BCNF

UNIT IV

Structured Query Language: Introduction, Commands in SQL, Data Types in SQL, SQL operators, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set operations, View, Sub Query.

UNIT V

PL/SQL: Introduction, Structure of PL/SQL program, PL/SQL Data Types, operators used in PL/SQL, variables, declaring variables in PL/SQL, Creating and running a PL/SQL Program, Control Structures: Conditional control statements, Iterative Control statements, Cursors: Types of cursors, Steps to create a Cursor, using cursors in PL/SQL program.

III. REFERENCES:

TEXT BOOKS:

1. Database management Systems, Alexis Leon and Mathews Leon, Vikas Publications 2002
2. Peter Rob, Carlos Coronel, Database Systems Design, Implementation and Management, Seventh Edition, Thomson (2007)
3. SQL, PL/SQL the Programming Language of Oracle, Ivan Bayross, BPB publications

REFERENCE BOOKS:

1. Elimasri / Navathe, Fundamentals of Database Systems, Fifth Edition, Pearson Addison Wesley (2007).
2. Database Principles, Programming, and Performance, P.O'Neil, E.O'Neil, 2nd ed., ELSEVIER.
3. SQL: The Ultimate Beginners Guide by Steve Tale.
4. Database System Concepts by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill
5. Database Management Systems by Raghu Ramakrishnan, McGrawhill

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz(on topics where the content can be compiled by smaller aspects and data (Individuals or groups a steams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured team activity)

III Semester
Course 2: Database Management Systems
Credits -1

1. Illustrate the creation of a table with constraints
2. Creation of college database and establish relationships between tables
3. Employee database

An enterprise wishes to maintain a database to automate its operations. Enterprise divided into certain departments and each department consists of employees. The following two tables describes the automation schemas.

Dept (deptno, dname, loc)

Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

Generate the following queries using data of above tables.

- i. List out all employees details
 - ii. Display empno, ename, job and sal columns of all employees
 - iii. Display employee details who are working as „CLERK“
 - iv. Find out number of employees working in each department
 - v. Find out job wise total salaries and number of employees.
 - vi. Calculate HRA as 30% and DA as 65% of salary
4. Demonstrate the use of GRANT and REVOKE commands to provide authorization

PL/SQL PROGRAMS

5. Write a PL/SQL program to check the given number is Armstrong or not.
 6. Write a PL/SQL program to check the given string is palindrome or not.
 7. Writ a PL/SQL program to generate multiplication tables
 8. Write a PL/SQL code to find the factorial of any number.
 9. Write a PL/SQL program to check the given number is palindrome or not.
 10. Write a PL/SQL program to display to 10 rows in Emp table based on their job and salary.
 11. Write a PL/SQL program to raise the employee salary by 10% for department number 30 people
- Write a procedure to update the salary of Employee, who are not getting commission by 10%

IV Semester
Course 3: Web Technologies
Credits -3

I. LEARNING OUTCOMES:

Students after successful completion of the course will be able to

1. To understand the web architecture and web services.
2. To practice latest web technologies and tools by conducting experiments.
3. To design interactive web pages using HTML and Style sheets.
4. To study the framework and building blocks of Integrated Development Environment.
5. To provide solutions by identifying and formulating IT related problems.

II. SYLLABUS:

UNIT I

Internet Language, Understanding HTML, HTML document structure, Create a Web Page, Publishing HTML Pages, Tags in HTML, title tag, Text Alignment tags, Text Formatting tags, heading tags, horizontal rule tag, paragraph tag, break tag. HTML Lists - Ordered List, Unordered List & Definition List – Using colors – Using Images

UNIT II

Horizontal Rule Tag - HTML Tables – Nested Tables - Hyperlinks: Textual, Graphical Links to sections – Multimedia Objects – Frames – Nested Frames – Forms – Form Controls: textbox, password, checkbox, radio button, select, text area - Processing of forms

UNIT III

JavaScript- Introduction, simple programming, Obtaining User Input with prompt Dialogs, Operators (arithmetic, Decision making, assignment, logical, increment and decrement) Control Structures - if... else selection statement, while, do... while repetitions statement, for statement, switch statement, break and continue statements.

UNIT IV

Functions – program modules in JavaScript, programmer defined functions, function definition, scope rules, global functions, and recursion JavaScript: Arrays, declaring arrays, accessing elements of an array.

UNIT V

Cascading Style Sheets: Introduction – Using Styles: As an attribute, tag & external file –Defining your own styles Properties and values: properties related to Fonts, Backgrounds & colors, text, boxes & borders Formatting blocks of information: Classes - Divisions – Spans - Layers with suitable examples.

III. REFERENCES:

TEXT BOOK:

1. Internet & World Wide Web - H.M.Deitel, P.J.Deitel, A.B.Goldberg-5th Edition

REFERENCE BOOKS

1. Programming Worldwide Web by RW Sebesta (Pearson)
2. An Introduction to Web Design + Programming by Wang & Katia (Pearson)
3. HTML & XML An Introduction NIIT(PHI)
4. HTML for the WWW with XHTML & CSS by Wlizabeth Castro (Pearson)
5. Fundamentals of the Internet and the World Wide Web by Raymond Green Law And Ellen Hepp (TMH)
6. Internet and Web Technologies by RajKamal(TMh)
7. Internet and WebBasics by NedSnell,BobTemple, TMClark(Pearson)

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Build a website with 10 pages for the case study identified.
2. Training of students by related industrial experts.
3. Assignments
4. Seminars, Group discussions, Quiz, Debates etc.(on related topics).

5. Presentation by students on best websites

IV Semester
Course 3: Web Technologies
Credits -1

V.WEB TECHNOLOGIES - PRACTICAL

1. Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.
2. Create your class timetable using table tag.
3. Create user Student feedback form (use textbox, text area, checkbox, radio button, select box etc.)
4. Write HTML code to develop a webpage having two frames that divide the webpage into two equal rows and then divide the row into equal columns fill each frame with a different background color.
5. Create your resume using HTML tags also experiment with colors, text , link ,size and also other tags you studied.
6. Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
7. Use Inline CSS to format your resume that you created.
8. Use External CSS to format your class timetable as you created.
9. Use External, Internal, and Inline CSS to format college web page that you created.
10. Develop a JavaScript to display today's date.
11. Develop simple calculator for addition, subtraction, and multiplication and division operation using JavaScript
12. Create HTML Page with JavaScript which takes Integer number as input and tells whether the number is ODD or EVEN.
13. Create HTML Page that contains form with fields Name, Email, Mobile No, Gender, Favorite Color and a button now write a JavaScript code to combine and display the information in textbox when the button is clicked

IV Semester
Course 4: Object Oriented Programming through Java
Credits -3

III. LEARNINGOUTCOMES:

Upon successful completion of the course, a student will be able to:

1. Understand the basic concepts of Object-Oriented Programming and Java Program Constructs
2. Implement classes and objects
2. Understand the benefits of code reusability achieved through inheritance
3. Demonstrate various classes in different packages and can design own packages
4. Learn the syntax and mechanisms of exception handling in Java
5. Learn how to create and manage threads and establish connections to database using JDBC.

II.SYLLABUS:

UNIT I

Introduction to Java: Features of Java, The Java virtual Machine, Structure of Java Program Naming Conventions and Data Types: Naming Conventions in Java, Data Types in Java, Literals Operators in Java: Operators

Control Statements in Java: if... else Statement, do... while Statement, while Loop, for Loop, for each loop, switch Statement , break Statement, continue Statement, return Statement Input and Output: Accepting

Input from the Keyboard: Reading Input with Scanner and Buffered Reader class, Displaying Output with System.out.println(), Displaying Formatted Output with String.format()

UNIT II

Arrays: Types of Arrays, array name.length, Command Line Arguments Strings: Creating Strings, String Class Methods. Introduction to OOPs: Problems in Procedure Oriented Approach, Features of Object-Oriented Programming System (OOPS) Classes and Objects: Object Creation, Initializing the Instance Variables, Access Specifiers, Constructors Methods in Java: Method Header or Method Prototype, Method Body, Understanding Methods, Static Methods, The keyword 'this', Instance Methods.

UNIT III

Inheritance: Inheritance, The keyword 'super', The Protected Specifier, Types of Inheritance Polymorphism: Polymorphism with Variables, Polymorphism using Methods, Polymorphism with Final Methods, final Class Type Casting: Casting Primitive Data Types, Casting Referenced Data Types, The Object Class Abstract Classes: Abstract Method and Abstract Class Interfaces: Interface, Multiple Inheritance using Interfaces Packages: Package, Different Types of Packages, Interfaces in a Package

UNIT – IV

Exception Handling: Errors in Java Program, Exceptions, throws Clause, throw Clause, Types of Exceptions, Re-throwing an Exception Streams: Stream, Creating a File using File Output Stream, Reading Data from a File using File Input Stream, Creating a File using File Writer, Reading a File using File Reader.

UNIT – V

Threads: Introduction, Thread Life Cycle, Creating a Thread and Running it, Terminating the Thread. Applets: Introduction, Creating an Applet, Uses of Applets, <APPLET> tag, A Simple Applet, Applet Parameters. Java Database Connectivity: Database Servers, Database Clients, JDBC (Java Database Connectivity), Working with Oracle Database, Stages in a JDBC Program,

III. REFERENCES

TEXT BOOKS:

1. CoreJava: An Integrated Approach, Authored by Dr. R. Nageswara Rao & Kogent Learning Solutions Inc.
2. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company
3. John R.Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TMH.
4. Deitel & Deitel. JavaTM: How to Program, PHI(2007)

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

2. Conduct coding competitions focused on object-oriented programming concepts in Java
3. Provide students with real-world scenarios and ask them to solve the given problems.
4. Assign group projects that require students to work together to create Java programs using OOP concepts

IV Semester

Course 4: Object Oriented Programming through Java

Credits -1

V. OBJECT ORIENTED PROGRAMMING THROUGH JAVA- PRACTICAL

2. Write a program to read Student Name, Reg.No, Marks [5] and calculate Total, Percentage, Result .Display all the details of students
3. Write a program to perform the following String Operations
 - a. Read a string
 - b. Find out whether there is a given sub string or not
 - c. Compare existing string by another string and display status

- d. Replace existing string character with another character
- e. Count number of characters in a string
4. Java program to implements Addition and Multiplication of two N X N matrices.
5. Java program to demonstrated use of Constructor.
6. Calculate area of the following shapes using method overloading.
 - a. Triangle
 - b. Rectangle
 - c. Circle
 - d. Square
7. Implement inheritance between Person (Aadhar, Surname, Name, DOB, and Age) and Student (Admission Number, College, Course, Year) classes where read Data(), display Data() are overriding methods.
8. Java program for implementing Interfaces
9. Java program on Multiple Inheritance.
10. Java program to display Serial Number from 1 to N by creating two Threads
11. Java program to demonstrate the following exception handlings
 - a. Divided by Zero
 - b. Array Index Out of Bound
 - c. Arithmetic Exception
 - d. User Defined Exception
12. Create an Applet to display different shapes such as Circle, Oval, Rectangle, Square and Triangle.
13. Write a program to create Book (ISBN, Title, Author, Price, Pages, Publisher) table and perform the following operations
 - a. Add book details
 - b. Search a book details for a given ISBN and display book details, if available
 - c. Update a book detail using ISBN
 - d. Delete book details for a given ISBN and display list of remaining Books

V Semester
Course 5: R Programming
 Credits -3

I. LEARNING OUTCOMES:

Upon successful completion of the course, a student will be able to:

1. Gain a solid understanding of R programming language
2. Acquire knowledge on various data structures and control structures in R.
3. Perform vectorized operations in R programming.
4. Develop skills in manipulating and transforming vectors, matrices, arrays and lists in R.
5. Explore and analyze data using data frames and tables.

II. SYLLABUS :

UNIT I

Introduction to R: R overview and history, Basic features of R, Benefits of R, data types in R, Installing R, Getting started with the RStudio IDE, Running R, Packages in R, variable names and assignment, operators, Input/output functions, reading and writing data.

UNIT II

Control structures: Conditional statements, Loops, dates and times, functions, String manipulations. Preview of Some Important R Data Structures: Vectors, Character Strings, Matrices, Lists, Data Frames, and Classes.

UNIT III

Vectors: Scalars, Vectors, Arrays and Matrices: Adding and Deleting Vector Elements, Obtaining the Length of a Vector Common vector operations: Arithmetic & logical operations, Vector Indexing, Generating vector sequences with seq(), Repeating vector constants with rep(), using all() and any() functions, Vectorized operations, NA and NULL values.

UNIT IV

Matrices and Arrays: Creating Matrices, General Matrix operations-linear algebra operations, matrix indexing, filtering on matrices, using apply() function, Add and Delete matrix rows and columns. Lists: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List, Accessing List Components and Values, Using lapply() and apply() functions.

UNIT-V

Data Frames: Creating Data Frames, Accessing Data Frames-Other Matrix-Like Operations: Extracting sub data frames, using rbind() and cbind() functions.

Factors and Tables : Factors and Levels - Common Functions Used with Factors : tapply(), split() and by() - Working with Tables, Matrix/Array-Like Operations on Tables, Extracting a Sub table- Math Functions: aggregate() and cut() functions.

III. REFERENCES :

TEXTBOOKS:

1. The Art of R Programming by Norman Matloff, Nostarch press, sanfransisco, 2011.
2. An Introduction to R for Beginners by SASHA HAFNER, on AUG-2019

REFERENCEBOOKS:

1. R Programming for Dummies, Andriette Vries and Joris Meys, Wiley
2. R for Data Science, Hadley Wickham, Garrett Grolemund, O'Reilly Media
3. R Programming: A Step-By-Step Guide for Absolute Beginners-2nd Edition, Daniel Daniel Bell
4. Learn R programming in 1 Day, Krishna Rungta, Published by Guru99

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assign students real-world data analysis projects that require them to apply their programming skills.
2. Organize coding challenges focused on R Programming.
3. Organize guest lectures or workshops.

V Semester

Course 5: R Programming

Credits -1

V.R PROGRAMMING - PRACTICAL

- 1) Write an R Program to take in put from user.
- 2) Write an R Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignment operators).
- 3) Write an R Program to Check if a Number is Odd or Even
- 4) Write an R Program to check if the given Number is a Prime Number
- 5) Write an R Program to Find the Factorial of a Number
- 6) Write an R Program to Find the Fibonacci sequence Using Recursive Function
- 7) Write an R Program to create a Vector and to access elements in a Vector
- 8) Write an R Program to create a Matrix from a Vector using dim() function.
- 9) Write an R Program to create a List and modify its components.

- 10) Write an R Program to create a Data Frame.
- 11) Write an R Program to access a Data Frame like a List.
- 12) Write an R Program to create a Factor

V Semester
Course 6: Software Engineering
Credits -3

I. LEARNING OUTCOMES:

Upon successful completion of the course, a student will be able to:

1. Understand and apply the fundamental principles of Object-Oriented Programming (OOP) concepts and Unified Modeling Language (UML) basics, in the development of software solutions.
2. Analyze and specify software requirements, develop use cases and scenarios, apply object-oriented analysis and design (OOAD) principles
3. Familiar with the concept of test-driven development (TDD) and its practical implementation
4. Analyze and Evaluate Software Maintenance and Evolution Strategies
5. Apply Advanced Object-Oriented Software Engineering Concepts.

II. SYLLABUS:

UNIT-I

Introduction to Object-Oriented Programming: Overview of software engineering, Introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, Introduction to software development process and software development lifecycle (SDLC)

UNIT-II

Requirements Analysis and Design: Requirements analysis and specification, Use cases and scenarios, Object-oriented analysis and design (OOAD), Design patterns, UML modelling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams)

UNIT-III

Software Construction and Testing: Software construction basics, Object-oriented design principles, Object-oriented programming languages (Java, C++, Python), Software testing basics (unit testing, integration testing, system testing), Test-driven development (TDD)

UNIT-IV

Software Maintenance and Evolution: Software maintenance basics, refactoring techniques, Software version control, Code review and inspection, Software evolution and reengineering

UNIT-V

Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), Aspect-oriented programming (AOP), Component-based software engineering (CBSE), Service-oriented architecture (SOA), Agile software development and Scrum methodologies.

III. REFERENCES:

TEXTBOOK(S)

1. An Introduction to Object Oriented Analysis and Design and the Unified Process, 3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

REFERENCE BOOKS

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville,1998, Wiley
2. Design Patterns, E.Gamma, R.Helm,R. Johnson,andJ.Vlissides
3. The Unified Modeling Language Reference Manual, J.Rumbaugh, I.Jacobson and G.Booch,Addison Wesley

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

1. Assign students real-world software development projects that require them to apply software engineering principles and practices.
2. Encourage students to prepare and deliver technical presentations or demonstrations on software engineering topics of their choice.
3. Invite industry professionals and experienced software engineers to deliver guest lectures or conduct workshops

V Semester
Course 6: Software Engineering
 Credits -1

V. SOFTWARE ENGINEERING - PRACTICAL

1. To perform the Requirement analysis of the specified problem and draw a flowchart
2. Understanding of System modeling: Data model i.e. ER – Diagram and draw The ER Diagram with generalization, specialization and aggregation of specified problem statement
3. Understanding of System modeling: Functional modeling: DFDlevel0i.e. Context Diagram and draw it
4. Understanding of System modeling: Functional modeling: DFDlevel1and DFDlevel2and draw it.
5. Understanding different actors and use cases in detail of the specified problem statement and draw it Using Rational Rose software any other available software.
6. To perform the user's view analysis: Use case diagram and draw it using Rational Rose or any other available software.
7. To draw the structural view diagram: Class diagram of specified problem statement using Rational Rose or any other available software.
8. To draw the behavioral view diagram: State-chart diagram, Activity diagram of specified problem Statement using Rational Rose any other available software.
9. TounderstandtestingandperformBoundaryvalueanalysisandEquivalenceclasstesting.
10. To draw Flow graph, DD paths , calculation of cyclomatic complexity and find out all the Independent paths from the D D paths graph.
11. Case study: Prepares RS for a given problem statement