d. Replace existing string character with another character
e. Count number of characters in a string
4. JavaprogramtoimplementsAdditionandMultiplicationoftwoNXNmatrices.
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. JavaprogramfortodisplaySerialNumberfrom1toNbycreating two Threads
11. Java program to demonstrate the following exception handlings
a. Divided by Zero
b. Array Index OutofBound
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)tableandperform 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. LEARNINGOUTCOMES:
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 :
UNITI
Introduction to R: R overview and history, Basic features of R, Benefits of R, data types inR, 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 l apply()and apply()functions.

UNIT-V

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

Factors and Tables : Factors and Levels - Common Functions Used with Factors : t apply() ,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. TheArt of R Programming by Norman Matlof, Nostarch press, sanfransisco,2011.
2. An Introduction to R for Beginners by SASHA HAFNER,on AUG-2019

REFERENCEBOOKS:

1. R Programming for Dummies, AndriedeVries and JorisMeys, Wiley
2. R for Data Science, Hadley Wickham, Garrett Grolemund ,O'Reilly Media
3. R Programming: AStep-By-StepGuideforAbsoluteBeginners-2ndEdition,DanielDanielBell
4. LearnRprogrammingin1Day, KrishnaRungta, PublishedbyGuru99

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

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. LEARNINGOUTCOMES:
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), Serviceoriented architecture (SOA), Agile software development and Scrum methodologies.
III. REFERENCES:
TEXTBOOK(S)
1. An Introduction to ObjectOrientedAnalysisandDesignandtheUnifiedProcess,3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press
REFERENCEBOOKS