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 objectoriented 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 1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville,1998, Wiley

2. Design Patterns, E.Gamma, R.Helm, R. Johnson, and J. Vlissides

3. The Unified Modeling Language Reference Manual, J.Rumbaugh, I.Jacobson and G.Boo ch,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.

 $9. \ To understand testing and perform Boundary value analysis and Equivalence class testing.$

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