

L	T	P	C
3	0	0	3

Course Code: CSE316

Semester: IV

SOFTWARE DESIGN WITH UML

Course Objectives:

This course will help the learner to understand the concepts of object-oriented technologies and unified modeling language to design the software in the aspects of interactions between use-cases, classes, packages, states, components, and execution architecture.

UNIT - I

11 Periods

Introduction. Software development process: The Waterfall Model vs. The Spiral Model. The Software Crisis, description of the real world using the Objects Model, Quality software characteristics. **Object oriented technologies:** Classes, inheritance and multiple configurations. Description of the Object- Oriented Analysis process vs. the Structure Analysis Model. The process of Object-oriented software development. Description of Design Patterns. Technological Description of Distributed Systems. **UML Language:** Standards - Elements of the language - General description of various models.

UNIT - II

12 Periods

Requirements Analysis Using Case Modeling. Analysis of system requirements. Actor definitions - Writing a case goal - Use Case Diagrams -Use Case Relationships.

Interaction diagrams (Transfer from Analysis to Design in the Characterization Stage). Description of goal - Defining UML Method, Operation, Object Interface, Class. **Sequence Diagram.** Finding objects from Flow of Events - Describing the process of finding objects using a Sequence Diagram - Describing the process of finding objects using a Collaboration Diagram.

UNIT - III

11 Periods

The Static Structure Diagrams (The Logical View Design Stage). The Class Diagram Model - Attributes descriptions - Operations descriptions - Connections descriptions in the Static Model. - Association, Generalization, Aggregation, Dependency, Interfacing, Multiplicity.

Package Diagram Model. Description of the model - White box, black box - Connections between packages – Interfaces - Create Package Diagram - Drill Down.

UNIT - IV

11 Periods

State Diagram / Activity Diagram (Dynamic Model). Description of the State Diagram - Events Handling - Description of the Activity Diagram - Exercise in State Machines.

Component Diagram Model. Physical Aspect - Logical Aspect - Connections and Dependencies - User face-Initial DB design in a UML environment.

Deployment Model. Processors – Connections – Components – Tasks – Threads - Signals and Events.

TEXTBOOKS

1. Bernd Bruegge and Allen H. Dutoit, *Object-Oriented Software Engineering: using UML, Patterns and Java*, Pearson publishers, 3rd Edition, 2011.
2. Martin Fowler, *UML Distilled*, Addison-Wesley Professional, 3rd edition, 2018.

REFERENCE

1. Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1st Edition, 1995.

UNITWISE LEARNING OUTCOMES

Upon successful completion of each unit, the learner will be able to

Unit I	<ul style="list-style-type: none">• Explain the software development process models and object-oriented technologies• Describe the standards and elements of the unified modelling language used for different models
Unit II	<ul style="list-style-type: none">• Prepare software requirements specification document for a given problem• Represent the relationships between the use-cases using diagrams• Design the interaction between the use-cases using sequence diagrams
Unit III	<ul style="list-style-type: none">• Depict the classes with attributes and operations and establish the relationship between them• Combine the classes as packages and design the interactions between the as a diagram
Unit IV	<ul style="list-style-type: none">• Model the system based on the activities and number of states• Draw component diagram and deployment diagram for the system under study