

OBJECT ORIENTED ANALYSIS AND DESIGN

V Semester: IT

Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACS009	Core	L	T	P	C	CIA	SEE	Total
		3	-	-	3	30	70	100
Contact Classes: 45	Tutorial Classes: Nil	Practical Classes: Nil			Total Classes: 45			

COURSE OBJECTIVES:
The course should enable the students to:

I. Develop the skills to analyze and design object-oriented problems.

II. Create design patterns to solve problems based on object oriented concepts.

III. Understand the various processes and techniques for building object-oriented software systems.

IV. Prepare unified modeling techniques for case studies.

COURSE OUTCOMES (COs):

CO 1: Discuss the overview of object oriented modeling and benefits of each.

CO 2: Differentiate advance object-oriented approach from the traditional approach for design and development of System.

CO 3: Understand Unified Modeling Language (UML) for representation of an object-oriented system using different modeling views.

CO 4: Apply appropriate design patterns to model or design of the system.

CO 5: Apply various software architectures, including frameworks and design patterns, when developing software projects.

COURSE LEARNING OUTCOMES (CLOs):

1. Able to show the importance of modeling concept for object oriented development in system.

2. Demonstrate the Conceptual model of UML and SDLC.

3. Able to understand the role and function of each UML model in software development using object-oriented approach.

4. Illustrate the importance of classes and their associated relationships by understanding various common mechanisms.

5. Able to differentiate advance object-oriented approach from the traditional approach for design and development of System.

6. Analyze the Objects and Classes are required for the development of software system.

7. Creation of interaction diagram that model the dynamic aspects of a software system.

8. Use case and activity studies to illustrate the analysis and design concepts.

9. Identify, analyze, and model behavioral concepts of the system and also know the importance of events and signals and their modeling techniques.

10. Analyze and understand the uses of process and threads and time and space to model and development of a system.

11. Demonstrate state machines and state chart diagrams and their modeling techniques.

12. Illustrate the uses of component and deployment diagram and their modeling techniques.

13. Understands how to apply the pattern based analysis and design to the software to be developed.

14. Describe how design patterns facilitate development and list several of the most popular patterns.

15. Identify and describe design patterns and their application in a software design project

16. An Ability to refactor poorly designed solutions by using the appropriate design patterns.

17. Develop UML models for design patterns using currently available software modeling tools. 18. Evaluate and apply design patterns, architectural patterns and enterprise patterns to the development of software systems. 19. Assess the use of Design patterns in the design of software systems and the refactoring of existing systems. 20. Analyze software components and case studies of system architecture and determine how integration with new and existing systems may be achieved		
UNIT-I	STRUCTURAL MODELLING	Classes: 10
Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, architecture, software development life cycle; Classes, relationships, common mechanisms and diagrams.		
UNIT-II	ADVANCED BEHAVIORAL MODELING	Classes: 08
Advanced classes, advanced relationships, interfaces, types and roles, packages, terms, concepts, modeling techniques for class and object diagrams; Interactions: Interaction diagrams; Use cases: Use case diagrams, activity diagrams.		
UNIT-III	ARCHITECTURAL MODELING	Classes: 08
Events and signals, state machines, processes and threads, time and space. State chart diagrams, component diagrams, deployment diagrams.		
UNIT-IV	DESIGN PATTERN	Classes: 09
GRASP: Designing objects with responsibilities, creator, information expert, low coupling, high cohesion, design patterns, creational, factory method, structural, bridge, adaptor, behavioral, strategy.		
UNIT-V	APPLYING DESIGN PATTENS	Classes: 10
System sequence diagrams, relation between sequence diagrams and use case logical architecture and UML package diagram, logical architecture refinement; Case study: The next gen POS system, inception, use case modeling, relating use cases, include, extend and generalization, domain models, domain model refinement.		
Text Books:		
1. Grady Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language User Guide”, Pearson Education, 2 nd Edition, 2004. 2. Craig Larman, “Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development”, Pearson Education, 3 rd Edition, 2005. 3. Enrich Gamma, Richard Helm, Ralph Johnson, John Vlissides, “Design Patterns”, Pearson Education, 2 nd Edition, 2009.		