OBJECT ORIENTED ANALYSIS AND DESIGN PATTERNS

VI Semester: CSE

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
ACS015	Core	3	1	3	4	30	70	100
Contact Classes: 45	Tutorial Classes: 15	Practical Classes: Nil				Total Classes: 60		

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: Understand Object Oriented and UML concepts
CO 2: Apply advanced behavioral modelling techniques in design and drawing UML diagrams for various systems
CO 3: Apply architectural modelling techniques in design and drawing UML diagrams for different systems
CO 4: Create design solutions for design problems by using ERASP and GOF patterns
CO 5: Apply design patterns for some case studies

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. Ability to refractor 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: 09**

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: 09**

Modeling techniques for class and object diagrams; Interactions: Interaction diagrams; Use cases: Use case diagrams, activity diagrams..

Unit -III	**ARCHITECTURAL MODELING**	**Classes: 09**

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, low coupling, high cohesion, design patterns, creational, factory method, structural, behavioral, strategy.

Unit -V	**APPLYING DESIGN PATTENS**	**Classes: 09**

System sequence diagrams, logical architecture refinement; domain models, domain model refinement Case study: The next gen POS system, inception.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Pearson Education, 2nd Edition, 2004.
2. Enrich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns", Pearson Education, 2nd Edition, 2009.
3. Craig Larman, "Applying UML and Patterns", 3rd Edition, 2011.

Reference Books:

1. Simon Bennett, Steve Mc Robb and Ray Farmer, "Object Oriented Systems Analysis and Design Using UML", McGraw-Hill Education, 4th Edition, 2010.
2. Pascal Roques, "Modeling Software Systems Using UML2", WILEY- Dreamtech India Pvt. Ltd, 2nd Edition, 2007.

Web References:

1. https://www.oodesign.com/
2. https://www.tutorialspoint.com/object_oriented_analysis_design/
3. https://www.startertutorials.com/uml/category/ooad

E-Text Books:
1. https://books-ith.cf/pdf/ebook-for-ooad-free-download-advanced-assembly-language-the-peter-norton-programming-library-0136587747-pdf-ibook.html.