OOAD AND DESIGN PATTERNS

	L	T	P	C
III B. Tech. - I Semester				
Course Code: A3CS22	3	1	-	3

COURSE OVERVIEW:

This course emphasises on Object Oriented software design and application of design patterns. Various types of design patterns are taught. Focus will be on Object Oriented Analysis of the system requirements followed by system design. This course helps in learning software design in a real world perspective.

COURSE OBJECTIVES:

1. To describe the object-oriented software development process, including object- oriented methodologies and work flow
2. To explain various UML diagrams

COURSE OUTCOMES:

At the end of the course student will be able to:

1. Analyze the requirements and generate use cases
2. Perform Object oriented analysis
3. Perform overall design using various UML diagrams

SYLLABUS

UNIT - I

INTRODUCTION TO UML: Introduction to object oriented concepts like inheritance, Polymorphism, Information hiding, Importance of modelling, Principles of modelling, Object oriented modelling, An overview of UML, Conceptual model of the UML, Architecture, Software development life cycle.

BASIC STRUCTURAL MODELING: *Classes:* Terms and concepts, Common modelling techniques; *Relationships* Modelling simple dependencies, Single inheritance and structural relationships; Common mechanisms and diagrams.

ADVANCED STRUCTURAL MODELING: Advance classes, Advance relationships, Interfaces, Types and Roles, Packages, Instances.

UNIT - II

THE OBJECT-ORIENTED DESIGN PROCESS: The object and class Concepts, Identifying classes, Identifying responsibilities, Relationships between Classes, Use Cases, CRC cards, UML class diagrams, Sequence diagrams, State diagrams, Using Java doc for design documentation, *Case Study:* A voice mail system.

UNIT - III

GUIDELINES FOR CLASS DESIGN: An overview of the date classes in the java library, designing a day class, the importance of encapsulation, analyzing the quality of an interface, programming by contract, unit testing.

INTERFACE TYPES AND POLYMORPHISM: The icon interface type, polymorphism, drawing shapes, the comparable interface type, the comparator interface type, anonymous classes, frames and user interface components, user interface actions, timers, designing an interface type.

UNIT - IV

PATTERNS AND GUI PROGRAMMING: Iterators, the pattern concept, the observer pattern, layout managers and the strategy pattern, components, containers and the composite pattern, scroll bars and the decorator pattern, how to recognize patterns, putting patterns to work.

INHERITANCE AND ABSTRACT CLASSES: The concept of inheritance, graphics programming with inheritance, abstract classes, the template method pattern, protected interfaces, the hierarchy of swing components, the hierarchy of standard geometric shapes, the hierarchy of exception classes, when not to use inheritance.

UNIT - V

FRAMEWORKS: Frameworks, applets as a simple framework, the collections framework, a graph editor framework, enhancing the graph editor framework.

MULTITHREADING: Thread basics, Thread synchronization, Animations.

MORE DESIGN PATTERNS: The Adapter pattern, Actions and the command pattern, the factory method pattern, the proxy pattern, the singleton pattern, the visitor pattern, other design patterns.

TEXT BOOKS:

1. Grady Booch, James Rumbaugh, Ivar Jacobson (2009), The Unified Modeling Language User guide, 2nd edition, Pearson Education, New Delhi, India.
2. Cay Horstmann (2004), Object-Oriented Design and Patterns, Wiley India edition, New Delhi, India.

REFERENCE BOOKS:

1. Meilir Page-Jones (2000), Fundamentals of Object Oriented Design in UML, Pearson Education and NewYork.
2. Craig Larman (2005), An introduction to Object –Oriented Analysis and Design and Unified Process Appling UML and Patterns, 3rdedition, Pearson Education, New Delhi, India.
3. John W. Satzinger, Robert B Jackson, Stephen D Burd (2004), Object-Oriented Analysis and Design with theUnified Process, Cengage learning, India.