OBJECT ORIENTED MODELING

Course Code: 13IT2103 **L P C**
4 0 3

Pre requisites: Software Engineering.

Course Educational Objectives:
The main objective of the course is to expose the students to model the software architecture using different UML diagrams, unified process. Upon completion of this course, the student should be able to:

1. Learn relationships.
2. Draw all UML diagrams.
3. Learn about forward engineering and reverse engineering.
4. Unified process and different phases of unified process.
5. Case studies any real time application using the unified process.

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

1. Represent a program, module even a system in the form of diagrams.
2. Get the knowledge of various modeling diagrams and their behaviors.
3. Analyze unified process and the 4P's of the process.
4. Analyze the Generic Iteration workflow.
5. Get the knowledge about various phases involved in the modeling.

UNIT- I
Introduction to UML: The meaning of Object Orientation, object identity, Encapsulation, information hiding, polymorphism, generosity, importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture.
Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.
Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT-II

Collaboration Diagrams: Terms, Concepts, depicting a message, iterated messages, use of self in messages.

Sequence Diagrams: Terms, concepts, depicting asynchronous messages with/without priority, callback mechanism, broadcast messages.

Basic Behavioral Modeling: Use cases, Use case Diagrams, Activity Diagrams.

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

UNIT-III

Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

The Unified process: use case driven, architecture centric, iterative, and incremental

The Four Ps: people, project, product, and process.

Use case driven process: why use case, capturing use cases, analysis, design, and implementation to realize the use cases, testing the use cases.

Architecture-centric process: architecture in brief, why we need architecture, use cases and architecture, the steps to architecture, an architecture description.

UNIT-IV

Iterative incremental process: iterative incremental in brief, why iterative incremental development? The iterative approach is risk driven, the generic iteration.

The Generic Iteration workflow: phases are the first division workflow, planning proceeds doing, risks affect project planning, use case prioritization, resource needed, assess the iteration and phases.

Inception phase: early in the inception phase, the archetypal inception iteration workflow, execute the core workflows, requirements to test.

UNIT-V

Elaboration Phase: elaboration phase in brief, early in the elaboration phase, the architectural elaboration iteration workflow, execute the core workflows-Requirements to test.

Construction phase: early in the construction phase, the archetypal construction iteration workflow, execute the core workflow.

Transition phase: early in the transition phase, activities in transition phase

Case Studies: Automation of a Library, Software Simulator application (2-floor elevator simulator)

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, *The Unified Modeling Language User Guide*, 2nd Edition, Pearson Education, 2007
2. Ivar Jacobson, Grady Booch, James Rumbaugh, *The Unified Software Development Process*, 1st Edition, Pearson Education, 2007.

References:

1. Meilir Page-Jones, *Fundamentals of Object Oriented Design in UML*, 1st Edition, Pearson Education, 2007.
2. Atul Kahate, *Object Oriented Analysis & Design*, 1st Edition, TMH, 2001.
3. Mark Priestley, *Practical Object-Oriented Design with UML*, 2nd Edition, TMH,2005.
4. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, *UML 2 Toolkit*, 1st Edition, WILEY-Dreamtech India Pvt. Ltd, 2003.

Web references:

1. http://modelica.org
2. http://openmodelica.org