OBJECT ORIENTED ANALYSIS AND DESIGN

SUB CODE: CE 405 / IT 405

Teaching Scheme (Credits and Hours)

Teaching scheme				Total Credit	Evaluation Scheme					
L	T	P	Total		Theory		Mid Sem Exam	CIA	Pract.	Total
Hrs	Hrs	Hrs	Hrs		Hrs	Marks	Marks	Marks	Marks	Marks
03	00	02	05	04	03	70	30	20	30	150

Learning Objectives:

The educational Objectives of this Course are:

- To Introduce various designing techniques and methods for object oriented
- Performance analysis with real time system
- Demonstrate a familiarity with object oriented data and system.
- To give clear idea on implementing design with UML diagram like state diagram , activity diagram , use case diagram etc.

Outline Of the Course:

Sr. No	Title of the Unit	Minimum Hours
1	Introduction	04
2	Modeling Concepts	05
3	Class Modeling	06
4	State Modeling	05
5	Interaction Modeling	04
6	Analysis and Design	07
7	System Design	07
8	Class design	07

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

Detailed Syllabus

Sr. No	Topic	Lecture Hours	Weight age(%)
1	**Introduction** • About Object Orientated Technology • Development and OO Modeling • History.	04	10
2	**Modeling Concepts** • Modeling design Technique • Three models, Class Model • State model • Interaction model	05	10
3	**Class Modeling** • Object and class concepts • link and association • Generalization • Inheritance • Advanced class modeling- aggregation • Abstract class • Metadata • Constraints	06	12
4	**State Modeling** • Event, state • Transition and conditions • state diagram • state diagram • behaviour • concurrency • Relation of Class • State models	05	12
5	**Interaction Modeling** • Use case Models • sequence models • activity models	04	12
6	**Analysis and Design** • Development Life cycle • Development stages • Domain Analysis-Domain • class model • domain state model • domain interaction model • Iterating and analysis. • Application Interaction model • Application class model • Application state Model • Adding operation	07	15

7	**System Design** • Estimating Performance • Making a reuse plan • breaking system into subsystems • identifying concurrency • allocation of subsystems, • management of data storage • Handling Global resources • choosing a software control strategy • Handling boundary condition • Common Architectural style	07	12
8	**Class design** • Overview of class design • designing algorithms recursing downward • refactoring • design optimization • Adjustment of Inheritance • Reification of Behaviour	07	12
	TOTAL	**45**	**100**

Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lecture and laboratory which carries 10 marks in overall evaluation.
- One internal exam will be conducted as a part of internal theory evaluation.
- Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation.
- Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation.
- The course includes a laboratory,where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments shall be performed in the laboratory related to course contents.

Students Learning Outcome:

On successful completion of the course, the student will:

- Develop modular solutions to a given problem statement,
- Design and implement software employing the principles of encapsulation, information hiding, abstraction, and polymorphism,
- Design, implement, and use classes and methods in an object-oriented programming language, employing standard naming conventions and making appropriate use of

advanced features such as inheritance, exception handling, I/O, references, and simple GUIs,

- Evaluate existing classes and software for the purposes of extension through inheritance,
- Use and create standard API documents to understand and document the use of classes and methods,
- Design and implement through test suites (unit testing),
- Refactor existing software to improve its design or efficiency,
- Use object-oriented design tools such as UML class diagrams to model problem solutions and express inheritance, association, aggregation, and composition relationships among classes,
- Recognize and use basic object-oriented design patterns to structure solutions to problems,
- Implement association relationships and multiplicities,
- Use frameworks, classes, and methods from standard libraries in problem solutions,
- Explain the fundamentals of software development including development process, quality of software systems, and challenges of software development, and
- Define or explain principles of modularity, encapsulation, information hiding, abstraction, and polymorphism

Reference Books:

1. Oriented Modeling and Design wih UML second edition by michael Blaha and James Rambaugh
2. Object-Oriented Analysis and Design with Applications, Second Edition
   by, Grady Booch

List of experiments:

Sr. No	Name of Experiment
1	E- R Diagram
2	Study of Class Diagram
3	Study of Use Case Diagram
4	Study of Sequence Diagram
5	Study of Activity Diagram
6	Study of package diagram
7	Study of State chart Diagram
8	Study of Collaboration Diagram
9	Study of Component Diagram
10	Study of Deployment Diagram