18UCSC601	Object Oriented Modeling and Design	(4-0-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs):
This is a 4 credit, 52 contact hours course at undergraduate level focusing on the process of object oriented system modeling, design, patterns and tools used in the industry to enable them to construct software system using various standards and techniques.

Course Outcomes (COs):

CO	**Description of the Course Outcome:** At the end of the course, the student will be able to:	Mapping to POs and PSOs		
		Substantial 3	**Moderate 2**	**Low 1**
CO-1	**Apply** fundamental Object Oriented concepts in solving problems.**[Usage][BL-3]**	13,14	1,2,3,15	-
CO-2	**Analyze** problem scenario and **identify** classes/ Objects, their properties and associations. **[Usage][BL-3/4]**	13	1,2,3,5,15	-
CO-3	**Analyze** problem scenario and **model** the system using UML diagrams. **[Usage][BL-3/4]**	13	1,2,3,5,15	-
CO-4	**Evaluate** the quality of Object Oriented system in-terms of Cohesion, coupling, sufficiency, completeness and primitiveness.**[Usage][BL-5]**	-	16	15
CO-5	**Implement** Object Oriented model in any Object Oriented language.**[Usage][BL-3]**	13,14	1,2,3	15
CO-6	**Identify** and **apply** the appropriate patterns in solving problems.**[Usage][BL-3]**	-	13,16	1
CO-7	**Propose** the appropriate strategies to incorporate standard quality parameters in the design of a system. **[Usage][BL-3]**	-	13,16	1

Note: BL- Bloom's Level

POs/ PSOs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mapping Level	1.7	2.0	2.0	-	2.0	-	-	-	-	-	-	-	2.7	3.0	1.6	2.0

Prerequisites: Knowledge of:

a. Any object oriented Programming Language.
b. Software Engineering.

Digitally signed by KURAKUNDI
RAGHAVENDRARAO
GOPINATH
Date: 2023-03-18 12:11+05:30

Course Contents

UNITS	Contents	Performance Ensuring Measures- PEM [Hours/ Course Outcomes/ Scope of Internal Assessments/ Assessment Tools/Evidences]				
		Durations in Hours		Course Outcomes	IAs	Assessment Tools
		Theory	Practice			
1	**Review:** Object Oriented Concepts and principles. **Introduction, modeling concepts, class modeling:** Object Orientation, developments themes; Evidence for usefulness of developments; modeling history. Modeling as Design Technique: Modeling; abstraction; The three models. Class Modeling: Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips.	10	-	1	1	1. **Class Test.** **(CO-1)** 2. **Written test-** CIE / SEE.**(CO-1)**
2	**Advanced class modeling, state modeling:** Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips. State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips. **Advanced state modeling, interaction modeling:** Advance state modeling: nested state diagrams; nested states; signal generalization; concurrency; a sample state model; relation of class and state models; practical tips. Interaction modeling: use case models; sequence models; activity models. Use case relationships; procedural sequence models; special constructs for	7	5	2 3 4	1 2	1. **Class test. (CO-2 &3)** 2. **Written test-** CIE/SEE.**(CO-2, 3)** 3. **CTA- Group Activity/ pair programming**: **Course project** Building reusable objects based on specification and writing test scripts based on OO model. **(CO-2,3)**

Digitally signed by KURAKUNDI
RAGHAVENDRARAO GOPINATH
Date: 2023-03-18 12:11+05:30

	activity models. **Evaluation**: OO system quality in terms of Cohesion, coupling, sufficiency, completeness and primitiveness. **Implementation** : OO design in appropriate language			5		
3	**Patterns–Part 1:** Introduction; layers, pipes and filters, blackboard.Distributed systems: broker; interactive systems: mvc, presentation-abstraction-control.	**6**	**4**	**6**	**2**	**1. Class test. (CO-6)** **2. Written test**- CIE/SEE. **(CO-6)** **3. CTA- Group Activity: Course project** on building system to demonstrate MVC/PAC **(CO-6)**
4	**Patterns—Part2:** Adaptable systems: microkernel; reflection. Structural decomposition**:** whole - part; organiz**a**tion of work: master - slave; access control: proxy. Others: Command Processor, View Handler, Forward Receiver, Client-Dispatcher-Server and publish Subscriber.	**7**	**3**	**6**	**3**	**1. Class test. (CO-6)** **2. Written test**- CIE/SEE. **(CO-6)** **3. CTA- Group Activity: Course project** on building system to demonstrate whole and part relationship **(CO-6)**
5	**Quality:** Functionality and architecture; architecture and quality attributes; system quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architecture qualities. Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Styles	**10**	**-**	**7**	**3**	**1. Written test**- CIE/SEE.**(CO-7)**
Other performance ensuring measures Like Industrial Visits, Course Projects, Implementation based assignments. Survey & Presentation, Certification, Conducting workshops/training programs, paper presentation, hobby projects, any engineering solutions for societal problems, Participation in relevant conference/training program/workshops etc…						**CTA:**Lecture /Training by **Industry expert** onUML tools and OO based Testing.**(CO- 2 & 3)** **CTA- Group Activity:** use of tools to prepare UML diagrams **(CO- 2 & 3)** Evaluation is to be based on report submission and oral presentation

Note: CTA: Course Teacher’s Assessment, CIE: Continuous Internal Evaluation, SEE: Semester End Examination.Procedure for conduction of IAs and SEE will be notified by the office of the Dean academic program and is common to all courses and programs.

Digitally signed by KURAKUNDI
RAGHAVENDRARAO GOPINATH
Date: 2023-03-18 12:11+05:30

Reference Books:

1) Michael Blaha, James Rumbaugh, “Object-Oriented Modeling and Design with UML”, 2/E, Pearson Education, 2007.
2) Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, “Pattern-Oriented Software Architecture”, A System of Patterns Volume 1, John Wiley and Sons, 2006.
3) Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice”, 2/E, Pearson Education, 2003.
4) Grady Boochetai, “Object-Oriented Analysis and Design with Applications”, 3/E, Pearson Education, 2007.
5) Ali Bahrami, “Object oriented systems development”, McGrawHill, 1999.
6) Mary Shaw and David Garlan, “Software Architecture Perspectives on an Emerging Discipline”, Prentice-Hall of India, 2007

Digitally signed by KURAKUNDI
RAGHAVENDRARAO GOPINATH
Date: 2023-03-18 12:11+05:30