

Subject Code: 2CS2010301	Subject Title: Software Engineering with UML
Pre-requisite	Systems & Object-Oriented Design Methodologies and Fundamentals of Structured Programming.

Course Objective:

The objectives of the course are to:

1. Get better Understand, Analyze and Model User's Requirements, Select Appropriate Process Model Apply it to All Stages of Software Development Life Cycle (SDLC)
2. Select and Apply Appropriate Metrics to Estimate Software Size, Effort, and Cost
3. Understand the Characteristics and Applicability of Various Software Tools
4. Able to teach the students tried-and-tested techniques widely embraced by experienced analysts plus new and emerging tools and techniques.
5. Get a good balanced exposure to both traditional and object-oriented approaches to system analysis & design.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	1	-	4	60	40	-	-	100

Subject Contents				
Sr. No	Topic	Total Hours	Weight (%)	
1	Introduction to Software Engineering & Process Models Software, Software Engineering, Software Characteristics, Software Application, Software Process, Layered Technology. Process Models – Waterfall, Incremental, Evolutionary Process Model– Prototype and Spiral Model. Agile Process; Extreme Programming (XP), Brief Overview of Other Agile Process Models: Adaptive Software Development, Scrum	12	25	
2	Principles of Software Engineering; and Requirements Modeling Introduction, Core Principles of Process and Practice, Principles Guiding Each Framework Activity. Requirements Engineering, Ground work for Understanding of Software Requirements, Overview of Eliciting Requirements, Building the requirements Model, Negotiating Requirements, Validating Requirements. Requirement Modeling Strategies, Overview of Flow-Oriented Modeling, Behavioral Modeling, Requirements Modeling for WebApp.	12	25	
3	Introduction to Object Oriented Analysis and Design Overview of UML, Conceptual Model of UML, Architecture, Software development life cycle Basic Structural Modeling Classes, Relationships, Class Diagrams	12	25	
4	Behavioral Modeling Interactions, Use Cases, Use Case Diagrams, Interaction Diagrams, Activity Diagrams Events and Signals, State Machines, State chart Diagram	12	25	

Course Outcome:

At the end of this course, the student would be able

- Design a solution to a given problem using one or more design patterns and implement the design in a programming language.
- Understand common lifecycle processes including waterfall (linear), incremental approaches (such as Unified process), and agile approaches.
- Learn Key modeling concepts that apply to both the traditional structured approach and the newer object- oriented approach
- Identify and analyze the system requirements using various system analysis techniques
- Design information system using structured and object-oriented techniques
- Model different views of information systems using object-oriented design patterns
- Recognize current and future trends of system analysis and design

List of References:

1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", 7th Edition, McGraw Hill Publications
2. Sommerville, "Software Engineering", 8th Edition, Pearson Education
3. Waman S. Jawadekar, "Software Engineering – Principles and Practices", TMGH Publication
4. Systems Analysis and Design by Kendall & Kendall, PHI Publication, 7th Edition
5. The Unified Modeling Language - User Guide by Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education Publication, 2009 Reprint
6. Object-Oriented Modeling and Design with UML by Michael Blaha, James Rumbaugh, Pearson Education Publication, 2nd Edition, 2007 Reprint
7. Object Oriented Analysis and Design Using UML by Mahesh P. Matha, PHI Publication
8. James Senn, Analysis and Design of Information system, Mc Graw Hill