

With effect from the A.Y 2018-19

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SYLLABUS FOR B.E IV SEMESTER**

**BASICS OF SOFTWARE ENGINEERING (Open Elective-II)
(for other Departments)**

Instruction: 1 Hr /week	SEE Marks :40	Course Code : OE510CS
Credits : 1	CIE Marks: 30	Duration of SEE : 2 Hrs

Course Objectives	Course Outcomes
Students should be able to	At the end of the course, students will be able to
<ul style="list-style-type: none">• Understand the concepts involved in the lifecycle of software development• Learn the best practices to be employed for the design and testing.	<ol style="list-style-type: none">1.Explain the various software development lifecycle models for a software system development.2.Build the prototype for software business case and analyze the requirements of software project.3.Analyze the different behavioral and structural models for the designed object oriented system.4.Identify verification and validation methods in a software engineering project and implement testing methods at various phases of SDLC

UNIT-I

Introduction to Software Engineering:

A generic view of Process: Software Engineering, Process Framework CMM Process Patterns, Process Assessment.

Process Models: Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Process.

An Agile view of Process: What is Agility, What is an Agile Process, Agile Process Models.

Requirements Engineering: A bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Negotiating Requirements, Validating Requirements.

UNIT-II

Object oriented Modeling & design using UML: Introduction to UML.

Structural Modeling: Classes and Advanced Classes, Relationships and Advanced Relationships, Common Mechanisms, Class Diagrams.

Behavioural Modelling: Interactions, Interaction diagrams, Use Cases, Use Case Diagrams, Activity diagrams, State Machines, State chart Diagrams.

Testing Tactics: Software testing fundamentals, Black box and White box testing.

Suggested Books:

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition (2005), Tata McGrawHill.
2. Grady Booch, James Rumbagu, Ivor Jacobson, The Unified Modeling Language-User guide, (Covering UML 2.0) ,2nd Edition (2007), Pearson Education, India.

Reference Books:

1. Shari Lawrence Pfleeger, Software engineering Theory and Practices, 4th Edition (2011), Pearson Education, India.
2. Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition (2005), Narosa Publishing House.

Online Resources:

1. <http://nptel.ac.in/courses/106101061/>
2. <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/>