

SOFTWARE ENGINEERING
(Effective from the academic year 2018 -2019)
SEMESTER – III

Course Code	18CS35	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03

CREDITS –3

Course Learning Objectives: This course (18CS35) will enable students to:

- Outline software engineering principles and activities involved in building large software programs. Identify ethical and professional issues and explain why they are of concern to software engineers.
- Explain the fundamentals of object oriented concepts
- Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation. Differentiate system models, use UML diagrams and apply design patterns.
- Discuss the distinctions between validation testing and defect testing.
- Recognize the importance of software maintenance and describe the intricacies involved in software evolution. Apply estimation techniques, schedule project activities and compute pricing.
- Identify software quality parameters and quantify software using measurements and metrics. List software quality standards and outline the practices involved.

Module 1

Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies.
Software Processes: Models: Waterfall Model (Sec 2.1.1), Incremental Model (Sec 2.1.2) and Spiral Model (Sec 2.1.3). Process activities.
Requirements Engineering: Requirements Engineering Processes (Chap 4). Requirements Elicitation and Analysis (Sec 4.5). Functional and non-functional requirements (Sec 4.1). The software Requirements Document (Sec 4.2). Requirements Specification (Sec 4.3). Requirements validation (Sec 4.6). Requirements Management (Sec 4.7).
RBT: L1, L2, L3

Contact Hours

08

Module 2

What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. **Introduction, Modelling Concepts and Class Modelling:** What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models;
Textbook 2: Ch 1,2,3.
RBT: L1, L2 L3

08

Module 3

System Models: Context models (Sec 5.1). Interaction models (Sec 5.2). Structural models (Sec 5.3). Behavioral models (Sec 5.4). Model-driven engineering (Sec 5.5).
Design and Implementation: Introduction to RUP (Sec 2.4), Design Principles (Chap 7). Object-oriented design using the UML (Sec 7.1). Design patterns (Sec 7.2). Implementation issues (Sec 7.3). Open source development (Sec 7.4).
RBT: L1, L2, L3

08

Module 4

Software Testing: Development testing (Sec 8.1), Test-driven development (Sec 8.2), Release testing (Sec 8.3), User testing (Sec 8.4), Test Automation (Page no 212). Software Evolution: Evolution processes (Sec 9.1), Program evolution dynamics (Sec 9.2), Software maintenance (Sec 9.3), Legacy system management (Sec 9.4). RBT: L1, L2, L3	08
Module 5	
Project Planning: Software pricing (Sec 23.1), Plan-driven development (Sec 23.2), Project scheduling (Sec 23.3), Estimation techniques (Sec 23.5). Quality management: Software quality (Sec 24.1), Reviews and inspections (Sec 24.3), Software measurement and metrics (Sec 24.4), Software standards (Sec 24.2) RBT: L1, L2, L3	08
Course Outcomes: The student will be able to : <ul style="list-style-type: none"> • Design a software system, component, or process to meet desired needs within realistic constraints. • Assess professional and ethical responsibility • Function on multi-disciplinary teams • Use the techniques, skills, and modern engineering tools necessary for engineering practice • Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems 	
Question Paper Pattern: <ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Textbooks: <ol style="list-style-type: none"> 1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24) 2. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML, 2nd Edition, Pearson Education, 2005. 	
Reference Books: <ol style="list-style-type: none"> 1. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill. 2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India 	

S. Chakraborty
H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engineering & Technology
Mijar, MOODBIDRI - 571225