Semester: 5	
Programme : B.Sc. Computer Science (Major)	
Course : SOFTWARE ENGINEERING	
Paper code: C3CS230522T / C3CS230522P	**Credits: 4**
Hours/week : Theory: 3, Practical: 2	
Category: Core/MDC/SEC/VAC : Core	
Theory / Practical / Composite : Composite	
No of Modules : 1	

Course Overview: This course introduces students to the principles and practices of Software Engineering, emphasizing systematic and disciplined approaches to software development. It covers the evolving role and characteristics of software, software process frameworks, and process models. The course addresses requirement engineering, software design principles, project management, cost estimation, and risk management. It also explores software testing fundamentals, testing levels, and testing types. Students gain knowledge of the complete software development lifecycle and acquire skills necessary to design, develop, test, and manage quality software systems in real-world environments.

Course Outcome:

1. **Recall and explain** fundamental concepts of software engineering, software processes, and lifecycle models.
2. **Apply** requirement analysis and modelling techniques to develop SRS documents.
3. **Apply** physical and data link layer concepts to analyse transmission media, multiplexing and switching techniques, encoding methods, framing, flow control, and error detection and correction mechanisms.
4. **Analyze** software design principles including coupling, cohesion, and architectural models.
5. **Design** software systems using structured methodologies and UML.
6. **Evaluate** project plans using estimation models such as Function Point and COCOMO and apply risk management strategies.
7. **Create** quality software solutions using testing principles, tools, and lifecycle practices.

Prerequisites:

SYLLABUS

UNIT/Module	CONTENT	HOURS or NUMBER OF CLASSES	CO Mapping	COGNITIVE LEVEL
I.	Introduction: Evolving role of software, Software characteristics, Software process framework, Framework & umbrella activities, Process models	6	**CO1**	K1, K2 (Remember, Explain)
II.	Requirement Analysis: Requirement engineering process, Analysis & modeling techniques, Flow-oriented modeling, SRS	8	**CO2, CO3**	K3, K4 (Apply, Analyze)

III.	Software Design: Design principles, Architecture, Module concepts, Coupling & cohesion, Structured design, Data design, UML	9	**CO3, CO4**	K4, K6 (Analyze, Create)
IV.	Software Project Management: Project phases, Estimation, Scheduling, Function Point, COCOMO, Risk management	8	**CO5**	K5 (Evaluate)
V.	Software Testing: Fundamentals, Levels of testing, Types of testing	8	**CO6**	K3, K6 (Apply, Create)

Text Books

1. Roger S. Pressman — Software Engineering: A Practitioner's Approach, McGraw Hill
2. Ian Sommerville — Software Engineering, Pearson
3. Pankaj Jalote — An Integrated Approach to Software Engineering, Narosa
4. Grady Booch — Object-Oriented Analysis and Design, Addison Wesley
5. Rajib Mall — Fundamentals of Software Engineering, PHI

Evaluation

Theory	**Practical**
CIA: 12 Attendance: 3 Semester Exam: 45	CA: 38 Attendance: 2

Paper Structure for Theory Semester Exam Module:
Answer 3 out of 5 questions of 15 marks each.

Course outcomes (COs) and Cognitive Level Mapping

COs	**CO Description**	**Cognitive levels**
CO1	Recall and explain SE fundamentals	K1-K2
CO2	Apply requirement analysis techniques	K3
CO3	Analyze design principles	K4
CO4	Design software systems using UML	K6
CO5	Evaluate project plans & risks	K5
C06	Create quality software using testing	K6