Course Title	**Software Engineering**		
Course Code	**CSH207B-T&P**		
Credits	Lecture:3		
Course type:	Lecture/Tutorial/Lab (3-1-2)		
Core or Elective:	Hard Core		
Term Offered:	Even Semester (B.Tech CSE-4th Semester)		
Course Schedule:	**Lecture:** 3 per week	**Tutorial:** 1 per week	**Lab:** 2 hours per week

Section-A

Introduction: Evolving role of software, Software Characteristics, Software crisis, Silver bullet, Software myths, Software process, Personal Software Process (PSP), Team Software Process (TSP), emergence of software engineering, project and product

Software Development Life-cycle: Software life cycle models: Build and fix, Waterfall, incremental and evolutionary process, model, spiral model, agile methodology, and selection of a life cycle model.

Section-B

Software Requirement Analysis and Specifications: Problem Analysis, Requirements engineering: Requirement elicitation, Requirement Analysis, Requirement documentation, Requirement Review, Types of requirements, Behavioral and non-behavioral requirement, feasibility study, Requirement elicitation: Interviews, Brainstorming, FAST, QFD, Use case Approach; Requirement analysis: DFDs, E-R diagram; Requirement documentation: Software Requirement Specification.

Software Project Planning: Project management concepts, planning the software project, Size Estimation—LOC based, FP based, COCOMO- A Heuristic estimation techniques, staffing level estimation, Putnam Resource Allocation model, risk analysis and management

Section-C

Software Design: Design concepts and principles: the design process, Modularity: Cohesion, Coupling, Strategies of design: bottom up, top down, hybrid design, User interface design, Object Oriented design: Class Diagrams, Interaction Diagrams-State chart Diagrams-Activity Diagrams

Software Metrics: Token Count,Data Structure Metrics, Information Flow Metrics.

Software Quality Models and Standards: Quality concepts, Software quality Assurance, SQA activities, CMM, The ISO 9000 Quality standards: The ISO approach to quality assurance systems, The ISO 9001 standard, software reliability

Section-D

Software Testing: Software process, Functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing: Path testing, Data flow and mutation testing, unit testing, integration and system testing, Debugging, Testing Tools & Standards.

Software Maintenance: Management of Maintenance, Maintenance Process, Maintenance Models, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation

LIST OF EXPERIMENTS: Tool Used: - Rational Rose Software

1. To identify the requirements of the project from the Problem statement and conduct Requirement elicitation techniques like Interviews, Brainstorming, FAST, QFD.
2. To perform E-R Modeling (E-R diagram, DFD) for the Problem Statement.
3. To Model UML Use Case Diagrams and capture Use Case Scenarios.
4. To design Software Requirement Specification document.
5. To estimate Size Metrics by calculating the number of Unadjusted Function points of the project using programming
6. To estimate Size Metrics by calculating the number of total Function point of the project using programming.
7. To draw Class Diagram and Interaction Diagrams.
8. To draw State Chart and Activity Diagrams.
9. To design test cases for the project using Black box testing.
10. To design test cases for the project using White box testing.

Text Books:

1. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 3rd ed., McGraw Hill Int. Ed., 1992.

2. K.K. Aggarwal&Yogesh Singh, “Software Engineering”, New Age International, 2001

Reference Books:

1. R. Fairley, “Software Engineering Concepts”, Tata McGraw Hill, 1997.

2. P. Jalote, “An Integrated approach to Software Engineering”, Narosa, 1991.

3. James Peter, W Pedrycz, “Software Engineering”, John Wiley & Sons

4. Sommerville, “Software Engineering”, Addison Wesley, 1999.