

## LESSON PLAN-SE

**Name of Faculty** : **Mr. Ravi Sachdeva**  
**Discipline** : **Computer Science Engg.**  
**Semester** : **B.Tech 6<sup>th</sup> Semester**  
**Subject** : **Software Engineering**  
**Lesson plan duration** : **15 Weeks**

**Work Load (Lecture/Practical) Per Week (in hours):** **Lecture - 03, Practical – 03**

| Week | Theory      |  | Practical     |   |
|------|-------------|--|---------------|---|
|      | Lecture day | Topic (Including assignment and test)  | Practical Day | Topic   |
| 1    | 1           | Introduction:<br>Introduction to Software Engineering                        | 1             | To identify the role of the software in today's world across a few significant domains related to day to day life |
|      | 2           | Software Characteristics, Software Crisis                                    |               |   |
|      | 3           | The Evolving role of Software, Software Development Life Cycle (SDLC) Models |               |   |
| 2    | 4           | Water Fall Model   | 2             | To identify the problem related to software crisis for a given scenario.  |
|      | 5           | Prototype Model  |               |   |
|      | 6           | Spiral Model   |               |   |
| 3    | 7           | Evolutionary Development Models  | 3             | To identify the problem related to software crisis for a given scenario-Contd.                                    |
|      | 8           | Iterative Enhancement Models   |               |   |
|      | 9           | RAD  |               |   |
| 4    | 10          | V Model  | 4             | To classify the requirement into functional and non-functional requirements.                                      |
|      | 11          | Class Test   |               |   |
|      | 12          | Software Requirement Specification:<br>Requirement Engineering Process       |               |   |
| 5    | 13          | Elicitation, Analysis, Documentation,  | 5             | To classify the requirement into functional and non-functional requirements-Contd.                                |
|      | 14          | Review and Management of User Needs, Feasibility Study                       |               |   |
|      | 15          | Data Flow Diagrams   |               |   |
| 6    | 16          | Decision Tables  | 6             | To implement at least four software   |

|    |    |  |    |   |
|----|----|--|----|---|
|    | 17 | SRS Document, IEEE Standard for SRS                              |    | metrics   |
|    | 18 | Software Quality, Concept of Software Quality Assurance (SQA)    |    |   |
| 7  | 19 | SEI-CMM Model  | 7  | To implement at least four software metrics-Contd.  |
|    | 20 | Introduction to Software Risk Management                         |    |   |
|    | 21 | Software Configuration Management                                |    |   |
| 8  | 22 | Class Test   | 8  | Preparation of requirement document for standard application problems in standard format.(e.g Library Management System, Railway Reservation system, Hospital management System, University Admission system) |
|    | 23 | Basic Concept of Software Design, Modularization                 |    |   |
|    | 24 | Design Structure Charts  |    |   |
| 9  | 25 | Pseudo Codes, Flow Charts, Coupling and Cohesion                 | 9  | Preparation of requirement document for standard application problems in standard format.(e.g Library Management System, Railway Reservation system, Hospital Management System, University Admission System) |
|    | 26 | Design Strategies: Function Oriented Design                      |    |   |
|    | 27 | Object Oriented Design, Top-Down and Bottom-Up Design.           |    |   |
| 10 | 28 | Software Measurement and Metrics: Various Size Oriented Measures | 10 | To prepare project schedule for standard application problems in standard format.   |
|    | 29 | Halstead's Software Science                                      |    |   |
|    | 30 | Function Point (FP) Based Measures                               |    |   |
| 11 | 31 | COCOMO, Cyclomatic Complexity Measures                           | 11 | To prepare project schedule for standard application problems in standard format – Contd.   |
|    | 32 | Control Flow Graphs.   |    |   |
|    | 33 | Class Test   |    |   |
| 12 | 34 | Software construction fundamentals, minimizing complexity        | 12 | To implement the functional testing techniques  |
|    | 35 | Top-Down and Bottom – Up programming,                            |    |   |

|    |    |   |    |  |
|----|----|---|----|--|
|    |    | structured programming                                |    |  |
|    | 36 | Compliance with Design and Coding Standards           |    |  |
| 13 | 37 | Testing Objectives, Unit Testing, Integration Testing | 13 | To implement the functional testing techniques-Contd.  |
|    | 38 | system testing, Acceptance Testing                    |    |  |
|    | 39 | Regression Testing, Structural Testing,               |    |  |
| 14 | 40 | Functional Testing                                    | 14 | To implement the structural testing techniques         |
|    | 41 | Debugging   |    |  |
|    | 42 | Maintenance: key issues                               |    |  |
| 15 | 43 | Types of software Maintenance, Cost of Maintenance    | 15 | To implement the structural testing techniques- Contd. |
|    | 44 | Software Re-Engineering                               |    |  |
|    | 45 | Class Test  |    |  |