

RAMAKRISHNA MISSION VIDYAMANDIRA

NEP Syllabus B.Sc. Computer Science Honours

Semester-V

Course Code: 5CMSMJC2

Course Type: Major Course

Course Outcome:

- Explain essential software engineering concepts and the software development life cycle.
- Apply software requirements accumulating techniques to develop functional and non-functional requirements.
- Analyse software design methods and select appropriate design patterns and architectures.
- Develop and test software components deploying suitable coding standards and testing methodologies.
- Evaluate software quality features and project management practices in real-world scenarios.

5CMSMJC2: Software Engineering

Credit: 3

Marks: 50

Introduction: Defining system, open and closed system, modeling of system through computer hardware, communication systems, external agents and software systems; Importance of Engineering Methodology towards computerization of a system. [3 L]

Software Life Cycle Models: Classical and Iterative Waterfall Model, Spiral Model, Prototyping Model, Evolutionary Model, Agile Model and their importance towards application for different system representations, Comparative Studies. [6 L]

Software Project Management: Project planning; Matrices for project size estimation: LOC, Function Point Metric; Heuristic and Empirical Modeling; COCOMO; Scheduling: Gantt chart, PERT chart; Analytical technique: Halstead's Software Science. [7 L]

Requirement Analysis and Specification: Requirements Principles and its analysis; Specification Principles and its representations; Need for SRS, Characteristics and components of SRS. [5 L]

Software Design: Cohesion and coupling; Function oriented design: Different levels of DFD Design, Physical and Logical DFD; Process Representation: Pseudo English, Tight English; Decision Tables and Trees, Structured analysis: Structure Chart Conversion from DFD; Object oriented design approach basics. [7 L]

Object Modeling: UML Overview; Use case model; Class diagram, Interaction diagram, State transition diagram; Activity diagram. [5 L]

Software Testing: Software Verification and Validation; Testing objectives, Testing Principles, Testability; Error and Faults; Unit Testing, White Box and Black Box Testing, Test Case Design: Test Vector, Test Stub. [7 L]

Software Quality Assurances: Concepts of Quality, Quality Control, Quality Assurance, SQA Activities, IEEE Standard for Statistical Software Quality Assurances (SSQA) criteria; Capability Maturity Model Integration. [5 L]

5CMSMJC2P: Application Software Development

Credit: 2

Marks: 25

- Object-oriented, event-driven, GUI application programming concepts.
- The Visual Integrated Development Environment.
- User interface design.
- Linking the program code with the interface.
- Writing and Debugging GUI programs; syntax errors, run-time errors, logic errors.
- Visual controls and user interface design, Variables and constants; types; scope and lifetime of variables and constants, Calculations and formatting of data, Decisions and conditions; selection statements, Procedures and Functions; parameters and arguments, Multiform projects; scope of variables and procedures; modules, Repetition statements.
- Connecting applications to database.

Recommended Books:

1. Fundamentals of Software Engineering by Rajib Mall, 3rd Edition; PHI.
 2. An Integrated Approach to Software Engineering by Pankaj Jalote, 3rd Edition; Narosa.
 3. Software Engineering by Pressman, 6th Edition; McGrawHill.
 4. Software Engineering by Ian Sommerville, 6th Edition; McGrawHill.
 5. The Unified Modeling Language User Guide, 2nd Edition; Pearson.
-