

MCA-14-13**SOFTWARE ENGINEERING****Maximum marks: 100 (External: 80, Internal: 20)****Time: 3 hours**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

UNIT - I

Introduction: Software Crisis–problem and causes, Software Processes, Software life cycle models: Waterfall, Prototype, Evolutionary and Spiral models, Overview of Quality Standards like ISO 9001, SEI-CMM, CMMI, PCMM, Six Sigma.

Software Metrics: Size Metrics like LOC, Token Count, Function Count, Design Metrics, Data Structure Metrics, Information Flow Metrics, cyclomatic complexity, Halstead Complexity measures.

UNIT - II

Software Project Planning: Cost estimation, static, Single and multivariate models, COCOMO model, Putnam Resource Allocation Model, Risk management, project scheduling, personnel planning, team structure, Software configuration management, quality assurance, project monitoring.

Software Requirement Analysis and Specifications: Structured Analysis, Data Flow Diagrams, Data Dictionaries, Entity–Relationship diagrams, Software Requirement and Specifications, Behavioral and non-behavioral requirements.

UNIT - III

Software Design: Design fundamentals, problem partitioning and abstraction, design methodology, Cohesion & Coupling, Function Oriented Design and User Interface Design.

Coding: Programming style, structured programming.

Software reliability: Metric and specification, Musa and JM reliability model, fault avoidance and tolerance, exception handling, defensive programming.

UNIT - IV

Software Testing: Functional testing: Boundary Value Analysis, Equivalence class testing, Cause effect graphing, Structural testing: Control flow based and data flow based testing, loop testing, mutation testing, load, stress and performance testing, software testing strategies: unit testing, integration testing, System testing, Alpha and Beta testing, debugging.

Static Testing: Formal Technical Reviews, Walk Through, Code Inspection.

Software Maintenance: Types of Maintenance, Maintenance Process, Maintenance characteristics, Reverse Engineering, Software Re-engineering.

Text Books:

1. Pressman R. S. , “Software Engineering - A practitioner’s approach”, Tata McGraw Hill.
2. Sommerville, “Software Engineering”, Pearson Education.

Reference Books:

1. Pfleeger, “Software Engineering: Theory and Practice”, Pearson Education.
2. P. Jalote, “An Integrated approach to Software Engineering”, Narosa Publications.
3. R. Fairley, “Software Engineering Concepts”, Tata McGraw Hill.
4. James Peter, W Pedrycz, “Software Engineering”, Wiley India Pvt. Ltd.