

1. A. Course Title Course Code Teaching Schemes

Course Title : **2CSDE80** - Software Testing and Quality Assurance

SEMESTER -VII

Year 2024-25

Course Code	Course Name	Teaching Scheme				Examination Scheme			
						Hours	Component Weightage		
		L	T	P	C	SEE	SEE	CE	LPW
2CSDE80	Software Testing and Quality Assurance	3	-	2	4	3	0.40	0.40	0.20

Course Outcomes:

At the end of the course, students will be able to -

1. interpret different types of testing techniques in depth
2. apply modern software testing strategies in relation to software development
3. design project test plans, test cases, test data to conduct test operations
4. develop practical skills related to software quality assurance

Syllabus**Teaching Hours: 45****Unit I****12**

Overview of Software Testing: Software Quality, Role of testing, testing approaches

Unit Testing: Concept of Unit Testing, Defect Prevention, Mutation Testing, Debugging, Unit Testing in eXtreme Programming

Control Flow Testing: Control Flow Graph, Paths in a Control Flow Graph, All-Path Coverage Criterion, Statement Coverage Criterion, Branch Coverage Criterion, Examples of Test Data Selection

Data Flow Testing: Data Flow Anomaly, Data Flow Graph, Data Flow Testing Criteria, Feasible Paths and Test Selection Criteria, Comparison of Testing Techniques.

Unit II**08**

System Integration Testing: System Integration Techniques, Types of Interfaces and Interface errors, Software and Hardware Integration, Off the shelf component testing, Built-in Testing

System Test Categories: Basic Tests, Functionality Tests, Robustness Tests, Interoperability Tests, Performance Tests, Scalability Tests, Stress Tests, Load and Stability Tests, Reliability Tests, Regression Tests, Documentation Tests

Functional Testing: Equivalence Class Partitioning Boundary value Analysis, Decision Tables, Random Testing, Error guessing, Category Partition

Unit III**08**

System Test Design and Planning: Test Design Factors, Requirement Identification, Characteristics of Testable Requirements, Test Design Preparedness Metrics, Test Case Design Effectiveness

System Test Planning, Automation: Structure of a System Test Plan, Beta Testing, System Test Automation

Unit IV**07**

System Test Execution: Metrics for tracking System Test, Beta Testing, System Test Report, Product Sustaining, Measuring Test Effectiveness

Acceptance Testing: Types of Acceptance Testing, Selection of Acceptance Criteria, Acceptance Test Execution, Acceptance Testing in eXtreme Programming.

Use cases for State-of-the-Art tools for carrying out Software Testing

Unit V**10**

Software Quality Assurance: Five views of Software Quality, McCall's Software Quality Factors, Quality Criteria, Relationship between Quality factors and Criteria, Components of SQA, Software Quality Standards and their requirements, Software Quality Metrics, Software Reliability Models

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings[^]:

1. Sagar Naik, Piyu Tripathy, Software Testing and Quality Assurance: Theory and Practice, Wiley.
2. William Perry, Effective Methods for Software Testing, Wiley.
3. Paul C. Jorgensen, Software Testing - A Craftsman's Approach, CRC Press.
4. Srinivasan Desikan and Gopalaswamy Ramesh, Software Testing, Pearson Education.
5. Louis Tamres, Introducing to Software Testing, Addison Wesley Publications.
6. Ron Patton, SAMS Techmedia Indian Edition, Software Testing, Pearson Education.
7. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons.
8. Robert V. Binder, Testing Object-Oriented Systems: Models Patterns and Tools, Addison Wesley.
9. Daniel Galin, Software Quality Assurance: From Theory to Implementation, Addison Wesley.
10. Stephen Kan, Metrics and Models in Software Quality Engineering, Addison Wesley.

L=Lecture, T=Tutorial, P=Practical, C=Credit

[^]this is not an exhaustive list.