COURSE CONTENT

SOFTWARE TESTING METHODOLOGIES

I Semester: CE / CSD / CSE / CSM / ECE / EEE / ME

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
24X0521	Foundation	3	0	0	3	40	60	100
Contact Classes: 45	Tutorial Classes: Nil	Practical Classes: Nil				Total Classes: 45		

Prerequisites: Software Engineering

Course Overview:

The course offers an in-depth exploration of software testing principles, techniques, and tools. It emphasizes both theoretical foundations and practical applications, covering various testing methodologies such as unit, integration, system, and object-oriented testing. Students will gain proficiency in designing test cases, understanding test processes, and utilizing automation tools to ensure software quality.

Course Objectives: The students will try to learn

1. Understand the fundamental concepts of software testing, including the purpose of testing, bug taxonomy, and the software testing life cycle.
2. Apply structural testing techniques such as flow graph-based path testing and data flow testing to detect software defects.
3. Analyze and implement domain, interface, and logic-based testing methods to verify software functionality and boundary conditions.
4. Design state-based and transition tests using state graphs and transition diagrams for dynamic system behavior testing.
5. Utilize graph theory and software testing tools (e.g., JMeter, WinRunner) to model software behavior and automate testing processes.

Course Outcomes: After Completion of the Course, Students should be able to

1. Identify different types of software bugs and apply suitable testing models and strategies.
2. Develop and execute flow graph, path-based, transaction, and dataflow test cases.
3. Perform domain and interface testing using boundary analysis and logic-based methods like decision tables and KV charts.
4. Design state transition tests and analyze software using graph matrices and node reduction techniques.
5. Use automated testing tools to implement real-world testing scenarios and improve software quality.

UNIT - I: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs, Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II: Transaction Flow Testing: - transaction flows, transaction flow testing techniques. Dataflow testing:- Basics of dataflow testing, strategies in dataflow testing, application of data flow testing.

UNIT - III: Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - IV: Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection. Logic Based Testing:- overview, decision tables, path expressions, kvcharts, specifications.

UNIT - V: State, State Graphs and Transition testing:- state graphs, good & bad state graphs,state testing, Testability tips.Graph Matrices and Application:-Motivational overview, matrix of graph, relations,power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner)

TEXT BOOKS:

1. Software Testing techniques – Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing – Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.

ELECTRONIC RESOURCES:

1. **https://market.tutorialspoint.com/ebook/agile-testing-tutorial/index.asp**
2. **https://market.tutorialspoint.com/ebook/software-testing-tutorial/index.asp**
3. **https://www.geeksforgeeks.org/software-testing/software-testing-tutorial/**

MATERIALS ONLINE:

1. Course template
2. Tutorial question bank
3. Tech talk and Concept Video topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. E-Learning Readiness Videos (ELRV)