SEMESTER S8

SOFTWARE TESTING

Course Code	**OECST833**	**CIE Marks**	40
Teaching Hours/Week (L: T:P: R)	3:0:0:0	**ESE Marks**	60
Credits	3	**Exam Hours**	2 Hrs. 30 Min.
Prerequisites (if any)	None	**Course Type**	Theory

Course Objectives:

1. To Cultivate proficiency in software testing methodologies and techniques.
2. To Foster expertise in software testing tools and technologies.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	**Introduction to Software Testing & Automation:-** Introduction to Software Testing - Concepts, importance of testing, software quality, and real-world failures (e.g., Ariane 5, Therac 25); Software Testing Processes - Levels of thinking in testing; Testing Terminologies - Verification, validation, fault, error, bug, test cases, and coverage criteria; Types of Testing - Unit, Integration, System, Acceptance, Performance (stress, usability, regression), and Security Testing; Industry Trends - AI in test case automation, Introduction to GenAI in testing; Testing Methods - Black-Box, White-Box, and Grey-Box Testing; Automation in Testing - Introduction to automation tools (e.g., Selenium, Cypress, JUnit); Case Study- Automation of Unit Testing and Mutation Testing using JUnit.	8
2	**Unit Testing, Mutation Testing & AI-Driven Automation:-** Unit Testing- Static and Dynamic Unit Testing, control flow testing, data flow testing, domain testing; Mutation Testing- Mutation operators, mutants, mutation score, and modern mutation testing tools (e.g., Muclipse); JUnit Framework - Automation of unit testing, frameworks for testing in real-world projects; AI in Testing - GenAI for test case generation and optimization, impact on automation; Industry Tools - Application of AI-driven testing tools in automation and predictive testing; Case Study - Mutation testing using JUnit, AI-enhanced test case automation.	8

3	**Advanced White Box Testing & Security Testing:-** Graph Coverage Criteria - Node, edge, and path coverage; prime path and round trip coverage; Data Flow Criteria - du paths, du pairs, subsumption relationships; Graph Coverage for Code - Control flow graphs (CFGs) for complex structures (e.g., loops, exceptions); Graph Coverage for Design Elements - Call graphs, class inheritance testing, and coupling data-flow pairs; Security Testing - Fundamentals, tools (OWASP, Burp Suite), and their role in protecting modern applications; Case Study - Application of graph based testing and security testing using industry standard tools.	**10**
4	**Black Box Testing, Grey Box Testing, and Responsive Testing:-** Black Box Testing - Input space partitioning, domain testing, functional testing (equivalence class partitioning, boundary value analysis, decision tables, random testing); Grey Box Testing - Introduction, advantages, and methodologies (matrix testing, regression testing, orthogonal array testing); Performance Testing - Network latency testing, browser compatibility, responsive testing across multiple devices (e.g., BrowserStack, LambdaTest); Introduction to PEX - Symbolic execution, parameterized unit testing, symbolic execution trees, and their application; GenAI in Testing - Advanced use cases for predictive and responsive testing across devices and environments; Case Study- Implementation of black-box, grey-box, and responsive testing using PEX and AI-driven tools.	**10**

Course Assessment Method
(CIE: 40 marks, ESE: 60 marks)

Continuous Internal Evaluation Marks (CIE):

Attendance	Assignment/ Microproject	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
5	**15**	**10**	**10**	**40**

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
• 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks **(8x3 =24 marks)**	• Each question carries 9 marks. • Two questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 3 subdivisions. **(4x9 = 36 marks)**	**60**

Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Demonstrate the ability to apply a range of software testing techniques, including unit testing using JUnit and automation tools.	**K2**
CO2	Illustrate using appropriate tools the mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods.	**K3**
CO3	Explain and apply graph coverage criteria in terms of control flow and data flow graphs to improve code quality.	**K2**
CO4	Demonstrate the importance of black-box approaches in terms of Domain and Functional Testing	**K3**
CO5	Illustrate the importance of security, compatibility, and performance testing across devices.	**K3**
CO6	Use advanced tools like PEX to perform symbolic execution and optimize test case generation and also leverage AI tools for automated test case prediction and symbolic execution with PEX.	**K3**

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3									3
CO2	3	3	3	3	3							3
CO3	3	3	3									3
CO4	3	3	3	3								3
CO5	3	3	3		3							3
CO6	3	3	3	3	3							3

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation*

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Introduction to Software Testing.	Paul Ammann, Jeff Offutt	Cambridge University Press	2/e, 2016
2	Software Testing and Quality Assurance: Theory and Practice	Kshirasagar Naik, Priyadarshi Tripathy	Wiley	1/e, 2008

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Software Testing	Ron Patten	Pearson	2/e, 2005
2	Software Testing: A Craftsman's Approach	Paul C. Jorgensen	CRC Press	4/e, 2017
3	Foundations of Software Testing	Dorothy Graham, Rex Black, Erik van Veenendaal	Cengage	4/e, 2021
4	The Art of Software Testing	Glenford J. Myers, Tom Badgett, Corey Sandler	Wiley	3/e, 2011

Video Links (NPTEL, SWAYAM…)	
Module No.	**Link ID**
1	https://archive.nptel.ac.in/courses/106/101/106101163/
2	https://archive.nptel.ac.in/courses/106/101/106101163/
3	https://archive.nptel.ac.in/courses/106/101/106101163/
4	https://archive.nptel.ac.in/courses/106/101/106101163/