UIT University

DEPARTMENT OF ENGINEERING TECHNOLOGY

(Spring 2025)

Course Information

Course Title: Software Testing Technologies		**Code:** SET221
Program: B.E(TECH)-Software	**Semester:** IV	**Credit Hours:** 1+1 **Lecture:** 16 **Practical:** 14
Knowledge Area (as per HEC curriculum template)	Computer Engineering Technology	

1. Course description and objectives:

The course introduces students to software testing concepts, techniques**,** and tools for ensuring software quality. Topics include Software Testing Life Cycle (STLC), Black Box & White Box Testing, Test Case Design, Defect Management, and Automated Testing Frameworks. Students will gain hands-on experience with Junit or Selenium or Google test and performance testing tools**.**

2. Course objectives:

By the end of the course, students will:

- Understand fundamental concepts of software testing, its models and types.
- Recognize different software testing techniques and select suitable testing approach based on project requirements.
- Develop and execute test cases using manual and automated testing tools.

3. Course Learning Outcomes (CLOs):

CLO No.	CLO Description	Domain and Taxonomy level	PLO mapped (i to xii)	Level of emphasis of the PLO (1=High; 2=Medium; 3=Low)
1.	Describe software testing and quality assurance model.	C1	i	2
2.	Recognize the suitable testing technique for a defined scenario.	C2	ii	2
3.	Apply the identified techniques and compile the findings of software testing activity.	P4	v	2
4.	Develop and execute test cases based on systematic testing strategies through automated testing tools.	P3	v	2

Note:

- ✓ *C → Cognitive, P → Psychomotor, A → Affective domains and 'n' is the taxonomy level.*
- ✓ *It is strongly suggested that one CLO should be mapped to one PLO and one domain only.*

Teaching Plan

4. Weekly Lecture Breakdown

Week #	Topic(s) to be covered	CLO #
1	**Introduction to Software Testing** What is Software Testing? Importance & objectives Software Testing vs Debugging Cost of defects in SDLC	CLO # 01
2	**Software Testing Life Cycle (STLC)** Phases: Requirement Analysis, Test Planning, Test Design, Test Execution, Closure Testing process models: Waterfall, Agile, V-Model	CLO # 01
3	**Testing Principles** **Test Level**	CLO # 01
4	**Types of Software Testing** Functional vs Non-functional Testing Static vs Dynamic Testing Manual vs Automated Testing	CLO # 02
5	**Static Testing Techniques** Reviews, Walkthroughs, Inspections, Static Analysis	CLO # 02
6	**Test Design Techniques: Introduction** Test Conditions, Designing Test Cases	CLO # 02
7	**Black Box Testing** Equivalence Partitioning Boundary Value Analysis	CLO # 02
8	**Black Box Testing** Decision Table Testing State Transition Testing Use Case Testing	CLO # 02
	MIDTERM EXAMINATIONS	
9	**White Box Testing** Control Flow Testing Code Coverage: Statement, Branch, Path	CLO # 02
10	**White Box Testing** Cyclomatic Complexity Mutation Testing	CLO # 02
11	**Test Management: Planning & Estimation** Test Strategies, Estimation Techniques	CLO # 02
12	Test Monitoring, Risk & Incident Management Risk-Based Testing, Test Metrics, Defect Tracking	CLO # 02
13	Configuration Management & Version Control Test Environment Setup, Change Management	CLO # 02

14	**Automated Testing** Introduction to Automation Testing Why Automate? Automation vs Manual Testing Test Automation Frameworks Types of Testing Tools, Benefits & Risks of Automation	CLO # 03
15	**Performance and SECURITY Testing** Load Testing Stress & Scalability Testing Security Testing	CLO # 02
16	**Bug Reporting and Defect Lifecycle** Defect Lifecycle (New, Assigned, Resolved, Closed) Bug Severity vs Priority Writing Effective Bug Reports Defect Tracking Tools	CLO # 02

Lab-work Plan

5. Experiment/Practical Breakdown

Experiment #	Experiment Title	CLO #
01	**Introduction to Testing Tools** Installation of Google Test and Visual studio\Eclipse.	4
02	**Writing and Executing Test cases** Test cases for sample Web/Desktop application	4
03	**Google Test Basics** Setting up Google Test for a C++ sample application.	3
04	**Writing Assertions in Google Test** Exploring various assertion types in Google Test. .	3
05	**Test Fixtures in Google Test** Using test fixtures to setup and teardown tests.	3
06	**Parameterized Testing with Google Test** Running the same test with different inputs.	3
07	**Open Ended Activity**	
08	**Installation of Selenium Basics** Setting up Selenium for web application testing.	
09	**Locating UI Elements with Selenium** Identifying elements on a webpage using different strategies.	3
10	**Interacting with UI Elements using Selenium** Performing actions on web elements.	3

11	**Handling Waits and Alerts in Selenium** Dealing with dynamic content and browser alerts	3
12	**Integrating Google Test with Selenium** Using Google Test assertions in Selenium tests	3
13	**Creating Test Suites with Google Test and Selenium** Organizing tests into suites for efficient execution	4
14	**Open Ended Activity**	3

6. **Syllabus and Books:**

Introduction to Software Testing: Importance, objectives, testing vs debugging, software quality assurance (QA) vs quality control (QC).Software Testing Life Cycle (STLC): Phases, test planning, test execution, test closure, entry and exit criteria.

Types of Testing: Functional vs Non-functional, Manual vs Automated, Regression Testing, Exploratory Testing. Black Box Testing Techniques: Equivalence Partitioning (EP), Boundary Value Analysis (BVA), Decision Table Testing, State Transition Testing. White Box Testing Techniques: Control Flow Testing, Code Coverage (Statement, Branch, Path), Cyclomatic Complexity**.** Test Case Development: Writing test cases, test scenario creation, test data preparation. Defect Lifecycle & Bug Reporting: Defect lifecycle (New, Assigned, Resolved, and Closed), Bug severity vs priority. Automated Testing Fundamentals: Why automation? Advantages & limitations, Test Automation Frameworks. Unit testing with xUnit Frameworks: Google Test, writing and executing unit test cases. I Testing & Record-and-Replay Tools: Selenium WebDriver, Cypress, automating UI interactions. Integration Testing: Stubs and drivers, API testing, Service virtualization. Performance Testing: Load testing, Stress testing, and Scalability testing. Security Testing: Common vulnerabilities, penetration testing, OWASP Top10.Test Management & CI/CD Integration: Automated test execution, Continuous Testing in CI/CD pipelines.

Text/Reference Books:

1. **Foundations of Software Testing ISTQB Certification**, Dorothy Graham, Rex Black, Erik Van Veenendaal, 4th Edition, ISBN 1473764793, Cengage Learning EMEA , 2019. 2.
2. **Introduction to Software Testing**, Paul Amman and Jeff Offut, Cambridge University Press; 2nd edition, 2016, ISBN: 978-1107172012 3.
3. **Complete Guide to Test Automation:** Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects , Arnon Axelrod, Apress, 2018, ISBN 978-1484238318

7. **Percentage of theoretical background, problems analysis and solution design**

Elements covered in the course	Percentage of full course coverage
Theoretical background	50%
Problem analysis	30%
Solution design	20%

8. Teaching and learning methods:

a. Lecture
b. Class discussion/ Videos
c. Presentation
d. Activities
e. Homework

9. Student assessment methods:

a. Quiz
b. Assignment
c. Exams (Theory)
d. Presentation
e. Project
f. Activities etc.

10. Assessment schedule:

a.	Quiz	throughout the semester	
b.	Assignment	throughout the semester	
c.	Exams		
d.	Midterm exam	Week	9
e.	Final theory exam	Week	18
f.	Presentation	throughout the semester	
g.	Activities	throughout the semester	

11. Weighting of assessments:

Theory:

a.	Quizzes/Activities	10 Marks
b.	Assignments/Presentation	20 Marks
c.	Midterm examination	20 Marks
d.	Final term examination	50 Marks
	Total	100 Marks

Lab:

a.	Sessional	20 Marks
b.	Final Lab examination/ viva	30 Marks
	Total	50 Marks

12. Facilities required for teaching and learning

a. Computer Usage
b. Software
c. Online board + online ppt writing
d. YouTube

Course group leader name: **Engr Bushra Aziz**

S. No.	Course group member (if any)	Theory/Lab	Signature
1	**Engr Bushra Aziz**	Theory	
2	**Engr Bushra Aziz**	Lab	

Recommended by the Program coordinator	Verified by the department Chairperson
Endorsed by the Dean of the faculty	**Approved by the Provost**

Program Learning Outcomes (Bachelor of Engineering Technology Program):

(i) **Engineering Technology Knowledge (SA1)**: An ability to apply knowledge of mathematics, natural science, Engineering Technology fundamentals and Engineering Technology specialization to defined and applied Engineering Technology procedures, processes, systems or methodologies.

(ii) **Problem Analysis (SA2)**: An ability to Identify, formulate, research literature and analyze broadly defined Engineering Technology problems reaching substantiated conclusions using analytical tools appropriate to the discipline or area of specialization.

(iii) **Design/Development of Solutions (SA3)**: An ability to design solutions for broadly-defined Engineering Technology problems and contribute to the design of systems, components or processes to meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

(iv) **Investigation (SA4)**: An ability to conduct investigations of broadly-defined problems; locate, search and select relevant data from codes, data bases and literature, design and conduct experiments to provide valid conclusions.

(v) **Modern Tool Usage (SA5)**: An ability to Select and apply appropriate techniques, resources, and modern technology and IT tools, including prediction and modelling, to broadly-defined Engineering Technology problems, with an understanding of the limitations.

(vi) **The Engineering Technologist and Society (SA6)**: An ability to demonstrate understanding of the societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to Engineering Technology practice and solutions to broadly defined Engineering Technology problems.

(vii) **Environment and Sustainability (SA7)**: An ability to understand and evaluate the sustainability and impact of Engineering Technology work in the solution of broadly defined Engineering Technology problems in societal and environmental contexts.

(viii) **Ethics (SA8)**: Understand and commit to professional ethics and responsibilities and norms of Engineering Technology practice

(ix) **Individual and Team Work (SA9)**: An ability to Function effectively as an individual, and as a member or leader in diverse teams.

(x) **Communication (SA10)**: An ability to communicate effectively on broadly defined Engineering Technology activities with the Engineering Technologist community and with society at large, by being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

(xi) **Project Management (SA11)**: An ability to demonstrate knowledge and understanding of Engineering Technology management principles and apply these to one's own work, as a member or leader in a team and to manage projects in multidisciplinary environments.

(xii) **Lifelong Learning (SA12)**: An ability to recognize the need for, and have the ability to engage in independent and life-long learning in specialist Engineering Technologies.