



**Gyanmanjari**  
Innovative University

Course Syllabus  
Gyanmanjari College of Computer Application  
Semester-6 (BCA)

**Subject:** Software Testing- BCAXX16330

**Type of course:** Major Core

**Prerequisite:** Software Engineering Basics, Basics of Java Programming

### Rationale:

Software testing is an indispensable process in the software development life cycle, serving as a critical quality assurance measure. Its primary objective is to identify and rectify defects, ensuring that the software meets specified requirements and functions as intended. Through a systematic and rigorous examination of the software, testing helps detect errors, inconsistencies, and potential vulnerabilities, thereby enhancing the reliability and robustness of the final product. By validating each component and feature, software testing contributes to the overall improvement of software quality, fosters user satisfaction, and reduces the likelihood of post-deployment issues. Additionally, it provides stakeholders with confidence in the software's performance, functionality, and security, ultimately supporting the delivery of a high-quality and dependable product.

### Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks	
CI	T	P		C	SEE		CCE			
					Theory	Practical	MSE	LWA		ALA
3	0	2	4	75	25	30	20	50	200	

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation, MSE- Mid Semester Examination; LWA - Lab Work Assessment; V – Viva voce; CCE-Continuous and Comprehensive Evaluation; ALA- Active Learning Activities.

3 Credits \* 25 Marks = 75 Marks (each credit carries 25 Marks) Theory

1 Credits \* 25 Marks = 25 Marks (each credit carries 25 Marks) Practical

SEE 100 Marks will be converted in to 50 Marks

CCE 100 Marks will be converted in to 50 Marks

It is compulsory to pass in each individual component.



### Course Content:

Sr. No	Course content	Hrs	% Weight age
1	<b>Introduction to Software Testing (Overview + New Concepts)</b> <b>Overview Only:</b> Software Engineering Concepts, SDLC & Models (Waterfall, Spiral, Incremental), Need & Importance of Software Testing, Role of Testing in Software Quality Assurance, Principles of Testing, Verification vs Validation, Error–Fault–Failure Relationship, Testing vs Debugging ,Software Testing Life Cycle (STLC) Overview	08	20%
2	<b>Test Planning &amp; Management</b> STLC Detailed – Requirement Analysis, Test Planning, Test Case Design ,Test Strategy, Test Plan Components, Entry & Exit Criteria, Test Environment Setup, Test Execution & Test Closure, Defect Life Cycle (DLC),Bug Tracking Tools Overview (Jira / Bugzilla)	10	20%
3	<b>Test Design Techniques</b> Test Case Design Techniques, Black Box: Equivalence Partitioning, BVA, Decision Table, State Transition, White Box: Statement, Branch, Condition, Path Coverage, Gray Box – Concept & Practical Relevance	08	20%
4	<b>Levels, Types &amp; Specialized Testing</b> Levels of Testing – Unit, Integration, System, Acceptance, Functional vs Non-Functional Testing, Specialized Testing: Regression, Smoke, Sanity, Compatibility, Localization, Usability, Security (Intro),Agile Testing – TDD, BDD, Role of Testers	11	20%
5	<b>Automation &amp; Emerging Trends</b> Need & Importance of Automation. Selenium WebDriver – Introduction ,Basics of JUnit / TestNG, Simple Automated Script (Demonstration), API Testing – Postman Overview + Sample, I & Testing in DevOps (GitHub Actions / Jenkins – Concepts),Exploratory & Ad-hoc Testing, Case Study – End-to-End Testing for E-commerce / Ticket Booking (Manual + Automation)	08	20%

### Continuous Assessment:

(For each activity maximum-minimum range is 5 to 10 marks)

Sr. No	Active Learning Activities	Marks
1	<b>QA Plan:</b> Students will assume the role of a QA/test manager and create a structured test plan document covering scope, objectives, strategy, environment, schedule, roles & responsibilities, deliverables and risks. The plan must reflect the project scenario provided and be submitted on the GMIU web portal.	10





2	<b>Code Path Testing:</b> Students will write a small program that includes decision logic (if/else, loops). Then they will apply white box testing techniques—statement coverage, branch coverage, and path coverage—to measure how completely the logic has been tested. Results must be documented and submitted on the GMIU web portal.	10
3	<b>Unit–Acceptance Chart:</b> Students will create a concise chart that outlines the four levels of software testing — Unit Testing, Integration Testing, System Testing, and Acceptance Testing — including brief definitions and simple examples for each level. They will then submit the chart on the GMIU web portal	10
4	<b>EP/BVA/Decision Table Cases:</b> Students will create test cases for a scenario using three black box techniques: Equivalence Partitioning (EP), Boundary Value Analysis (BVA), and Decision Table Testing. They will identify input partitions, boundary values, and decision combinations, then select representative test values and document their expected outcomes. The completed work is to be uploaded to the GMIU web portal.	10
5	<b>Mini-Project: Selenium &amp; API Testing Project</b> Students will work in groups of 4–5 members to create and execute automated test scripts using Selenium WebDriver and JUnit/TestNG, perform API testing using Postman, and integrate tests into a CI pipeline. They will document their scripts, results, and screenshots, and submit the report on the GMIU web portal.	10
Total		50

### Suggested Specification table with Marks (Theory):75

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weight age	25%	45%	15%	15%	0	0

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

### Course Outcome:

After learning the course the students should be able to:	
CO1	Explain the importance of software testing & the role of testing in software quality assurance.
CO2	Apply test planning, management & design techniques to build structured test cases
CO3	Compare and perform various levels, types & specialized testing approaches including Agile, TDD & BDD.
CO4	Demonstrate automation testing using Selenium, JUnit/TestNG & integrate automated tests



	in CI pipelines
CO5	Perform end-to-end test execution manually & using automation tools on real-world case studies.

### List of Practical

Sr. No	Descriptions	Unit No	Hrs
1	Introduction to Software Testing & Installation of Selenium	1	2
2	Case Study – How testing improves software quality; prepare short report	1	2
3	Test Case Design using EP / BVA / Decision Table	2	2
4	Create a Test Plan / Test Strategy for a sample application	2	2
5	Explore Jira/Bugzilla & log sample defects with lifecycle stages.	2	2
6	Apply Boundary Value Analysis for a given module.	3	2
7	To examine and measure code coverage by performing statement, branch, and path testing on a given program.	3	2
8	Demonstrate Levels of Testing (unit, integration, system, and acceptance) on sample project.	4	2
9	Prepare functional vs non-functional test scenarios for web/mobile application.	4	2
10	Perform Regression, Smoke & Sanity testing on an application.	4	2
11	Write Selenium Web Driver script to open website & fetch title.	5	2
12	API Testing using Postman – Overview & Sample Test	5	2
13	Perform API testing using Postman – create request & validate response.	5	2
14	Execute automated test using JUnit / TestNG	5	2
15	Case Study – End-to-End testing (Manual + Automation) on e-commerce/ticket booking system.	5	2

### Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, E-courses, Virtual Laboratory.





The internal evaluation will be done on the basis of Active Learning Assignment.  
Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

**Reference Books:**

- [1] Software Engineering: A Practitioner's Approach – 2019 – Roger S. Pressman, Bruce R. Maxim – McGraw Hill Education
- [2] Modern Software Engineering – 2021 – Rajiv Chopra – Khanna Publishing House
- [3] Software Testing and Quality Assurance: Theory and Practice – 2008 – Kshirasagar Naik, Priyadarshi Tripathy – Wiley India
- [4] Foundations of Software Testing – 2015 – Rex Black, Erik van Veenendaal, Dorothy Graham – Cengage Learning
- [5] Software Quality Assurance: From Theory to Implementation – 2001 – Daniel Galin – Pearson Education

