FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications

- **Sem.** : 6
- **Subject Code** : 05BC3601
- **Subject** : Software Testing
- **Course Objectives** :

1. To understand the basics of software testing and its processes.
2. To understand and apply various testing techniques.
3. To understand and appraise levels of testing.
4. To prepare test plan, design and various test reports.
5. To understand software quality management and its models.

- **Prerequisites:** Knowledge of System Development Life Cycle

Unit No	Topics Covered	No of lectures required
1	**Introduction to Software Testing and Software Testing Terminology and Methodology:** Introduction, Software Testing Definition, Goals of Software Testing, Software Testing As Process, Failure, Fault, Error, Test Case, Testware, Incident, Test Oracle, Life cycle of Bug, States of Bug, Bugs Classification based on criticality, Bugs Classification based on SDLC, Testing Principle, STLC, Verification and Validation Activities	**10**
2	**Testing Techniques** Dynamic Testing: Black-Box Testing Techniques, Boundary Value Analysis, Equivalence Class Testing, State Table Based Testing, Decision Table Based Testing, Cause-Effect Graphing Based Testing, Error Guessing. Dynamic Testing : White-Box Testing Techniques, Need of White-box testing, Logic Coverage Criteria, Basis Path Testing, Graph Matrices, Loop testing, Data Flow testing, Mutation testing	**10**

	Static Techniques Introduction, Benefit of static testing, Types of Static techniques, Inspections, Inspection Team, Inspection Process, Benefit of Inspection Process, Effectiveness of Inspection Process, Structured Walkthroughs, Technical Review	
3	**Validation Activities** Unit Validation Testing, Benefit of Designing stub and driver, Integration Testing, Decomposition based Integration, Comparison between top down and bottom up integration testing, Call Graph Based Integration, Pair-wise Integration, Neighborhood Integration, Path Based Integration, Function Testing, System Testing, Categories of System Tests, Acceptance Testing, Alpha Testing, Beta Testing, Regression Testing.	**10**
4	**Test Management** Test Organization, Structure of Test Group, Test Planning, Test Plan Components, Test Plan Hierarchy, Master Test Plan, Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log, Test Incident Report, Test Summary Report	**10**
5	**Software Quality Management** Software Quality, Broadening the concept of Quality, Quality cost, Benefits of Investment on Quality, Quality Control and Quality Assurance, Quality Management QM, QM and Project Management, Quality Factors, Methods of Quality Management, Procedural Approach to QM, Quantitative Approach to QM, Software Quality Metrics, SQA Models, ISO 9126, Capability Maturity Model CMM, Software Total Quality Management STQM	**10**

Course Outcomes :

1. Student will be able to understand the difference between bugs, errors and faults and build basic test cases.
2. Student will be able to apply testing techniques.
3. Student will be able to compare various levels of testing.
4. Student will be able to prepare test plans, design and various test reports.
5. Student will be able to understand software quality and its various models.

FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications

Course Outcomes – Program Outcomes Mapping Table :

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PSO1	PSO2	PSO3
CO1	H	M	-	-	L	-	H	M	-	L	-
CO2	H	H	-	-	L	-	H	M	H	M	L
CO3	H	M	-	-	L	-	H	M	H	M	L
CO4	H	M	L	M	L	-	H	M	H	M	-
CO5	H	L	-	M	L	-	H	M	-	-	-

Text Book :

1. **Software Testing Principles and Practices, Naresh Chauhan, Oxford University Press, Second Edition**

Reference Books :

1. **Software Testing Concepts and Tools, Nageshwar Rao Pusuluri, Dreamtech Press, First Edition**
2. **Software Testing: Concepts and Practices, K. Mustafa and R.A. Khan, Narosa, First Edition**
3. **Software Testing: Principles, Techniques and Tools, M G Limaye, McGraw Hill Education, First Edition**

Web References :

1. **www.istqb.org**
2. **www.softwaretestinghelp.com**

App References :

1. **Software Testing | QA Learning (NK Mobile Education)**
2. **Software Testing (Asha Tech Solutions)**

Syllabus Coverage from text /reference book & web/app reference:

Unit #	Chapter Numbers
1	1,2,3
2	4,5,6
3	7,8
4	9
5	13

FACULTY OF COMPUTER APPLICATIONS
Bachelor of Computer Applications

PRACTICALS

Unit No	List of Practicals
1 to 5	**Introduction to Test Cases:** Task: Write test cases for a simple login form. Solution: Test case 1: Enter valid username and password, expect successful login. Test case 2: Enter invalid username and password, expect login failure. Test case 3: Leave username or password field empty, expect appropriate error message. **Boundary Value Analysis:** Task-1: Test a function that calculates the discount based on purchase amount. Solution: Test case 1: Test with the minimum purchase amount. Test case 2: Test with a typical purchase amount. Test case 3: Test with the maximum purchase amount. **Boundary Value Analysis (Date Input):** Task-2: Test a system that processes date inputs. Solution: Test cases for the minimum and maximum valid dates, and dates just above and below acceptable ranges. **Equivalence Partitioning:** Task-1: Test a function that categorizes users into different age groups. Solution: Test case 1: Test with an age below the lower limit of a category. Test case 2: Test with an age within the valid range of a category. Test case 3: Test with an age above the upper limit of a category. **Equivalence Partitioning (Currency Conversion):** Task-2: Test a currency conversion function. Solution: Create test cases for currencies within the same category (equivalent) and from different categories (non-equivalent). **Error Guessing:** Task: Identify potential defects in a login system and write test cases to expose them. Solution: Test case 1: Attempt to log in with a very long username. Test case 2: Attempt to log in with SQL injection characters in the password. **Pairwise Testing:** Task: Test a form with multiple input fields for various combinations. Solution:

	Identify pairs of input values and create test cases to cover all possible combinations efficiently. **Security Testing:** Task: Test a login system for common security vulnerabilities. Solution: Test case 1: Attempt to log in with a SQL injection. Test case 2: Check for session timeout after a period of inactivity. **Performance Testing:** Task: Test the response time of a web page. Solution: Use browser developer tools to measure the load time of the page and identify potential performance issues. **Compatibility Testing:** Task: Test a web application on different browsers. Solution: Test the application on popular browsers like Chrome, Firefox, and Edge to ensure compatibility. **Volume Testing:** Task: Test the system with a large volume of data. Solution: Populate a database with a significant amount of data and assess the system's performance and responsiveness. **Recovery Testing:** Task: Simulate system failures and assess recovery mechanisms. Solution: Test scenarios where servers crash, and databases become unavailable to verify the system's recovery procedures. **User Acceptance Testing (UAT):** Task: Create test cases for end-users to validate the system's overall functionality. Solution: Engage end-users to execute predefined test cases and provide feedback on the system's usability. **Mutation Testing:** Task: Introduce intentional bugs into the code and create test cases to detect them. Solution: Introduce simple bugs (e.g., change arithmetic operators) and verify if the test suite detects them. **You can use additional other examples to perform all above testing.**