Lab 1: Fundamentals of Software Testing

Objective:

To understand the core concepts of software testing including test levels, test types, and black box testing techniques, and to gain hands-on experience designing black box test cases.

Theory Overview:

1. Fundamentals of Software Testing
    - Software Testing is the process of evaluating software to identify errors, gaps, or unmet requirements.
    - Key Concepts:
        - Error: Human mistake
        - Fault (Bug): Manifestation of an error in code
        - Failure: System deviation due to a fault
    - Static vs. Dynamic Testing:
        - Static: Without executing code (e.g., reviews)
        - Dynamic: With code execution (e.g., running test cases)
    - Verification vs. Validation:
        - Verification: Are we building the product right?
        - Validation: Are we building the right product?
2. Test Levels
    - Unit Testing: Testing individual components (e.g., functions or classes).
    - Integration Testing: Testing the interaction between integrated modules.
    - System Testing: Testing the complete and integrated application.
    - Acceptance Testing: Validating the system with respect to business requirements.
3. Test Types
    - Functional Testing: Tests features against requirements (e.g., login validation).
    - Non-functional Testing: Checks performance, usability, security, etc.
    - Regression Testing: Verifying changes didn't break existing functionality.
    - Smoke Testing: Initial testing to check system stability.
    - Sanity Testing: Focused testing after small changes.
4. Black Box Testing
    - Testing without knowledge of internal code.

- Focuses on inputs and expected outputs.

5. Basic Black Box Testing Techniques

- Equivalence Partitioning: Dividing input data into valid/invalid partitions.
- Boundary Value Analysis (BVA): Testing edges of input ranges.
- Decision Table Testing: Tabular representation of input combinations and outcomes.
- State Transition Testing: Validates behavior based on state changes.

Lab Tasks:

- Task 1: Understanding Test Levels
  - Consider a calculator app. Identify what you would test in:
    - Unit testing
    - Integration testing
    - System testing
    - Acceptance testing
- Task 2: Identify Test Types

  Given a login module:
  - Write 2 test cases for functional testing.
  - Mention 1 possible non-functional test (e.g., login speed).
  - Identify a scenario for regression and smoke testing.
- Task 3: Black Box Test Case Design

  Scenario: Test a "Student Grading" system:
  - Input: Marks (0–100)
  - Output: Grade (A/B/C/F)

  Design black box test cases using:
  - Equivalence Partitioning
    - Valid: 0–100
    - Invalid: <0, >100
  - Boundary Value Analysis
    - Test boundaries like 0, 1, 99, 100
  - Decision Table
    - Example:

Marks	Grade
90–100	A

70–89	B
50–69	C
<50	F

- State Transition
    - Imagine a Login System with states: Logged Out → Logging In → Logged In → Session Timeout → Logged Out
    - Draw a state transition diagram and write one test case for each transition.

Expected Outcomes:

- Demonstrate understanding of test levels and types.
- Apply black box testing techniques effectively.
- Create structured test cases for simple applications.

Lab Task Submission Format:

- Title Page
- Objective
- Theory Summary
- Tasks with Answers/Test Cases
- Conclusion

Conclusion:

This lab introduces foundational knowledge and practical skills in software testing. By the end of this session, students should understand how to design and classify test cases based on testing principles and black box techniques.