

# Practical No. 8 – Designing Test Suites (Software Testing)

**Aim:** To design test cases and test suites for verifying the functionality of a software system using software testing techniques such as unit testing and integration testing.

**Introduction:** Software testing is a process used to identify the correctness, completeness, and quality of developed software. It helps ensure that the system performs as expected and meets user requirements. Testing detects defects early, improves product quality, and enhances reliability before deployment.

**Objectives:**

- Understand the need for software testing.
- Learn about different software testing techniques and frameworks.
- Design test cases and test suites to verify software functionality.
- Implement unit and integration testing to identify and fix bugs.

**Theory:**

**Software Testing:** Software Testing is the process of executing a program to detect errors and ensure that the system behaves as expected.

**Verification and Validation:** Verification ensures the software meets design specifications, while Validation ensures it meets user needs.

**Standards for Software Test Documentation:** The IEEE 829 standard defines documentation for testing, including test plans, design, and reports.

**Testing Frameworks:** Frameworks like JUnit, pytest, and Selenium provide structured environments for managing test cases.

**Need for Software Testing:** Detect and fix bugs early, improve reliability, ensure performance, and meet customer satisfaction.

**Test Cases and Test Suite:** A Test Case defines inputs, conditions, and expected outputs, while a Test Suite groups related test cases.

**Types of Testing:** Unit Testing (individual functions), Integration Testing (module interaction), System Testing (complete system), and Acceptance Testing (business requirements).

**Sample Test Cases:**

Case ID	Test Scenario	Input Data	Expected Output	Actual Output	Status
TC_01	Verify login functionality	Username = 'user1', Password = '12345'	Login successful	Login successful	Pass
TC_02	Verify product selection process	Select product list	Product displayed	Product displayed	Pass
TC_03	Verify price comparison	Product A = ₹500, Product B = ₹450	Select lowest (B)	B selected	Pass
TC_04	Verify payment calculation	Amount = ₹1000, Discount = 10%	Final = ₹900	₹900	Pass
TC_05	Verify receipt generation	Payment done	Receipt generated	Receipt generated	Pass

**Example – Unit Testing (JUnit Example in Java):**

```
import static org.junit.Assert.*;
import org.junit.Test;
public class PaymentTest {
    @Test
    public void testDiscountCalculation() {
        double amount = 1000;
        double discount = 0.10;
        double expected = 900;
```

```
double actual = amount - (amount * discount); assertEquals(expected, actual, 0.001); } }
```

**Conclusion:** Software testing ensures that the developed application meets all requirements and functions correctly. Designing structured test cases and test suites helps detect bugs early, improves software quality, and verifies module integration through unit and integration testing.