Exercise 1 – Setting Up JUnit in a Java Project

# 1. Objective

To set up JUnit in a Java project and create a basic unit test using either Maven or manual setup.

# 2. Problem Statement / Scenario

You are beginning Java development and need to set up unit testing using JUnit. The goal is to create a Java project, add JUnit as a test framework, and write your first test class.

# 3. Approach / Steps

## 3.1 Understanding JUnit

JUnit is a popular testing framework for Java that allows developers to write and run repeatable tests. JUnit 4 uses annotations like @Test to define unit test methods.

## 3.2 Setup

You can set up JUnit using an IDE like IntelliJ IDEA or Eclipse, or manually using Maven or downloaded JARs.

## 3.3 Implementation

Create a Java class and a test class in the project. Use assertEquals in test methods to verify expected outputs.

# 4. Code

Add the following to your pom.xml inside <dependencies>:

<dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
</dependency>

Java Class: Calculator.java

package com.test;  
  
public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

JUnit Test Class: CalculatorTest.java

package com.test;  
  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 assertEquals(5, calc.add(2, 3));  
 }  
}

# 5. Output Verification

After running the test, the IDE or Maven will show output like:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0  
BUILD SUCCESS

# 6. Conclusion

JUnit is a lightweight and effective unit testing tool for Java applications. By writing test cases, developers can validate their code logic and catch bugs early in the development process.