**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit**

Scenario: You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

**1**.**Verified environment:**

Junit\_Testing/

├── src/

│ ├── main/java/com/example/Calculator.java

│ └── test/java/com/example/CalculatorTestWithFixtures.java

├── pom.xml

If src/test/java doesn’t exist, create it manually and set it as a Test Package Folder in Project Properties → Sources.

**2. Code for Class Under Test: Calculator.java**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**3. Code for Test Class: CalculatorTestWithFixtures.java**

Location: src/test/java/com/example/CalculatorTestWithFixtures.java

package com.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTestWithFixtures {

private Calculator calculator;

// Setup method - runs before each test

@Before

public void setUp() {

calculator = new Calculator();

System.out.println("Setup: Calculator instance created.");

}

// Teardown method - runs after each test

@After

public void tearDown() {

calculator = null;

System.out.println("Teardown: Calculator instance cleared.");

}

@Test

public void testAddition() {

int result = calculator.add(10, 5);

assertEquals(15, result);

}

@Test

public void testSubtraction() {

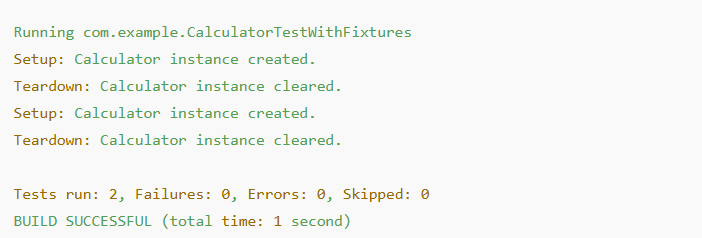
int result = calculator.subtract(10, 4);

assertEquals(6, result);

}

}

**OUTPUT OF THE CODE:**

****