**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

* Used the previously configured Maven project with JUnit 4.13.2 dependency.
* Confirmed structure and test package path:  
  src/test/java/com/example

**2.** Created 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.** Created Test Class: CalculatorTestWithFixtures.java

**Location**: src/test/java/com/example

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() {

// Arrange

int a = 10;

int b = 5;

// Act

int result = calculator.add(a, b);

// Assert

assertEquals(15, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

// Act

int result = calculator.subtract(a, b);

// Assert

assertEquals(6, result);

}

}

**Final Output:**

