Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and

Teardown Methods in JUnit

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calc;

@Before

public void setUp() {

System.out.println("Initializing Calculator instance...");

calc = new Calculator();

}

@After

public void tearDown() {

System.out.println("Destroying Calculator instance...\n");

calc = null;

}

@Test

public void testAddition() {

int a = 3;

int b = 7;

int expected = 10;

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

assertEquals("Addition result mismatch!", expected, result);

}

@Test

public void testSubtraction() {

int a = 15;

int b = 5;

int expected = 10;

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

assertEquals("Subtraction result mismatch!", expected, result);

}

@Test

public void testDivision() {

int a = 25;

int b = 5;

int expected = 5;

int result = calc.divide(a, b);

assertEquals("Division result mismatch!", expected, result);

}

@Test(expected = IllegalArgumentException.class)

public void testDivisionByZero() {

calc.divide(10, 0);

}

}

class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int divide(int a, int b) {

if (b == 0) {

throw new IllegalArgumentException("Cannot divide by zero");

}

return a / b;

}

}

**Output:**

