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("Setting up...");

calc = new Calculator();

}

@After

public void tearDown() {

System.out.println("Cleaning up...\n");

calc = null;

}

@Test

public void testAddition() {

int a = 4;

int b = 6;

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

assertEquals(10, result);

}

@Test

public void testSubtraction() {

int a = 10;

int b = 3;

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

assertEquals(7, result);

}

@Test

public void testDivision() {

int a = 20;

int b = 5;

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

assertEquals(4, result);

}

}

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:**

