**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.

Step-1 : ‘CalculatorFixtureTest’ Test is written using the AAA pattern and @Before and @After annotations are also used for setup and teardown methods.

Code (Main):

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int divide(int a, int b) {

if (b == 0) throw new IllegalArgumentException("Division by zero");

return a / b;

}

}

Code (Test):

package com.example;

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

public class CalculatorFixtureTest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator();

System.***out***.println("Setup: Creating Calculator object");

}

*@After*

public void tearDown() {

calculator = null;

System.***out***.println("Teardown: Destroying Calculator object");

}

*@Test*

public void testAddition() {

// Act

int result = calculator.add(4, 6);

// Assert

*assertEquals*(10, result);

}

*@Test*

public void testDivision() {

// Act

int result = calculator.divide(10, 2);

// Assert

*assertEquals*(5, result);

}

*@Test*(expected = IllegalArgumentException.class)

public void testDivideByZero() {

// Act

calculator.divide(10, 0);

}

}

Output:

A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.