**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

**Scenario:**

You need to set up JUnit in your Java project to start writing unit tests.

**Steps:**

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).

2. Add JUnit dependency to your project. If you are using Maven, add the following to your

pom.xml:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

3. Create a new test class in your project.

**Program:**

**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**Calculator.java**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**CalculatorTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalculatorTest {

*@Test*

public void testAdd() {

Calculator calc = new Calculator();

*assertEquals*(5, calc.add(2, 3));

*assertEquals*(0, calc.add(-7, 7));

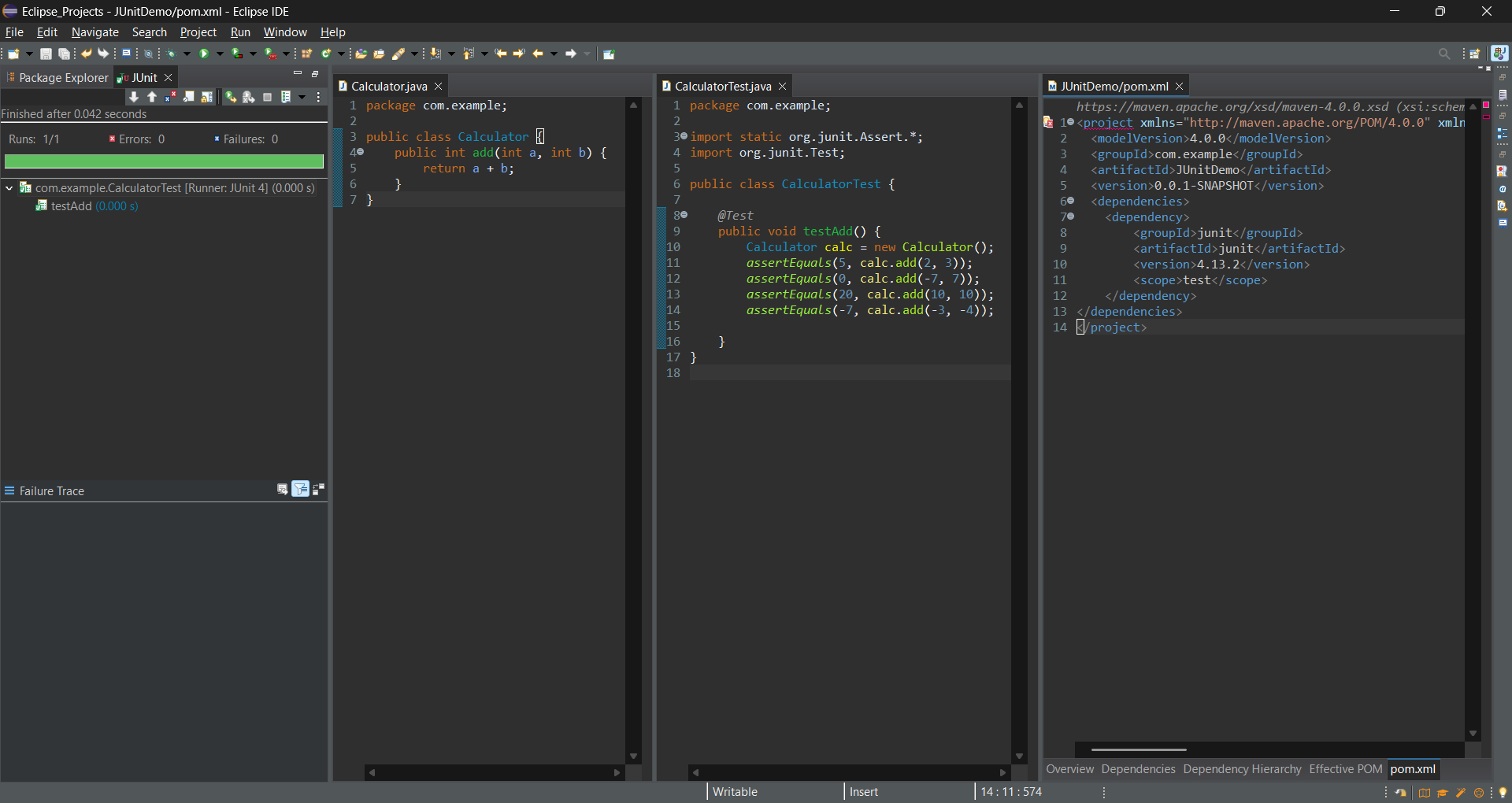
*assertEquals*(20, calc.add(10, 10));

*assertEquals*(-7, calc.add(-3, -4));

}

}

Output:



**Exercise 2: Writing Basic JUnit Tests**

**Scenario:**

You need to write basic JUnit tests for a simple Java class.

**Steps:**

1. Create a new Java class with some methods to test.

2. Write JUnit tests for these methods.

**Program:**

**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**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;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) {

throw new ArithmeticException("Division by zero is not allowed.");

}

return a / b;

}

}

**CalculatorTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalculatorTest {

Calculator calculator = new Calculator();

*@Test*

public void testAdd() {

*assertEquals*(5, calculator.add(2, 3));

*assertEquals*(0, calculator.add(-2, 2));

}

*@Test*

public void testSubtract() {

*assertEquals*(1, calculator.subtract(3, 2));

*assertEquals*(-5, calculator.subtract(0, 5));

}

*@Test*

public void testMultiply() {

*assertEquals*(6, calculator.multiply(2, 3));

*assertEquals*(0, calculator.multiply(0, 5));

}

*@Test*

public void testDivide() {

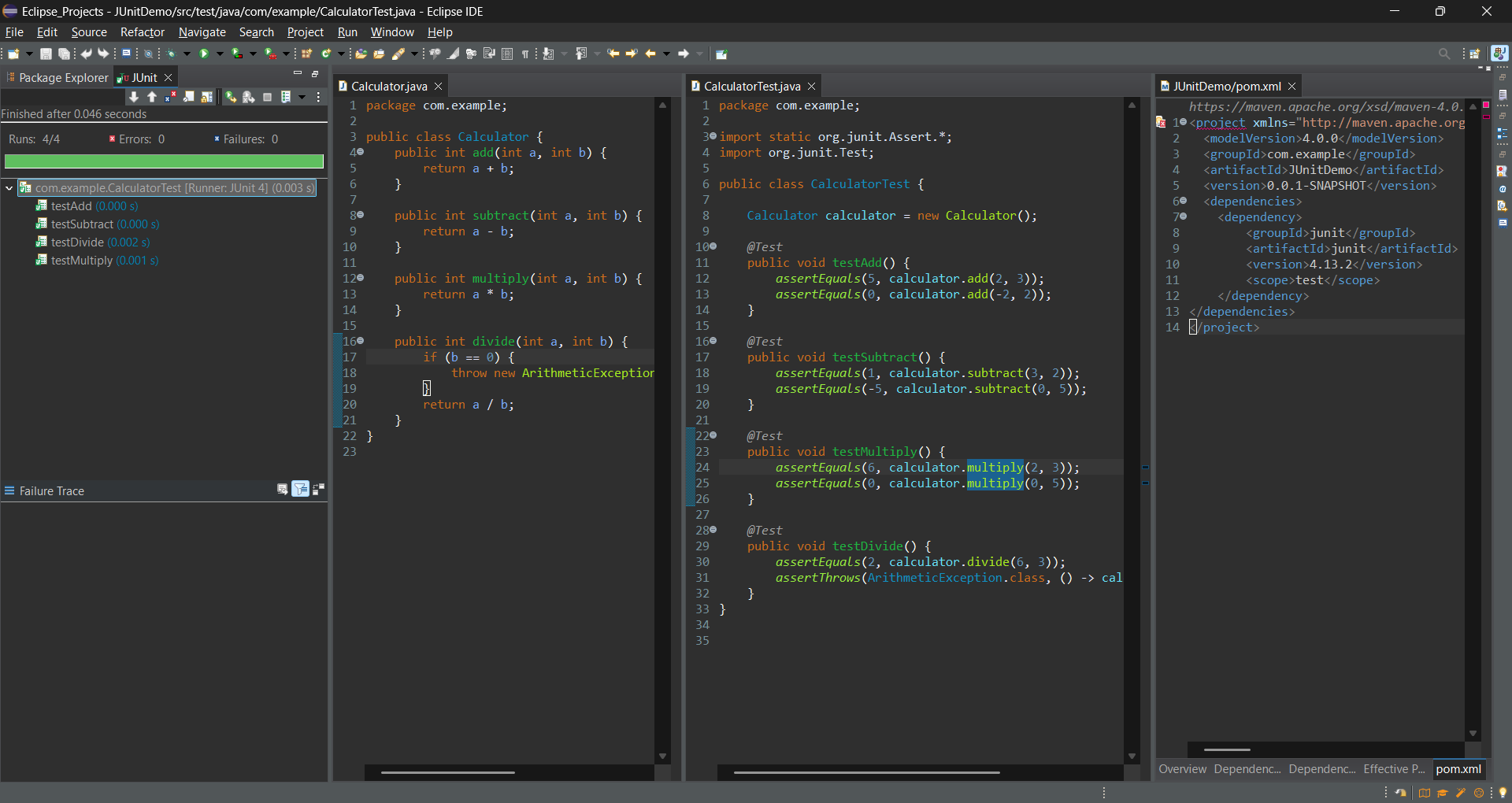
*assertEquals*(2, calculator.divide(6, 3));

*assertThrows*(ArithmeticException.class, () -> calculator.divide(10, 0));

}

}

**Output:**



**Exercise 3: Assertions in JUnit**

**Scenario:**

You need to use different assertions in JUnit to validate your test results.

**Steps:**

1. Write tests using various JUnit assertions.

Solution Code:

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert equals

assertEquals(5, 2 + 3);

// Assert true

assertTrue(5 > 3);

// Assert false

assertFalse(5 < 3);

// Assert null

assertNull(null);

// Assert not null

assertNotNull(new Object());

}

}

Program:

Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>AssertionsPractice</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

AssertionsTest.java

package com.example;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class AssertionsTest {

*@Test*

public void testAssertions() {

*assertEquals*(5, 2 + 3, "2 + 3 should equal 5");

*assertNotEquals*(4, 2 + 3, "2 + 3 should not equal 4");

*assertTrue*(5 > 3, "5 is greater than 3");

*assertFalse*(5 < 3, "5 is not less than 3");

String str = null;

*assertNull*(str, "String should be null");

String nonNullStr = "JUnit";

*assertNotNull*(nonNullStr, "String should not be null");

String a = "hello";

String b = a;

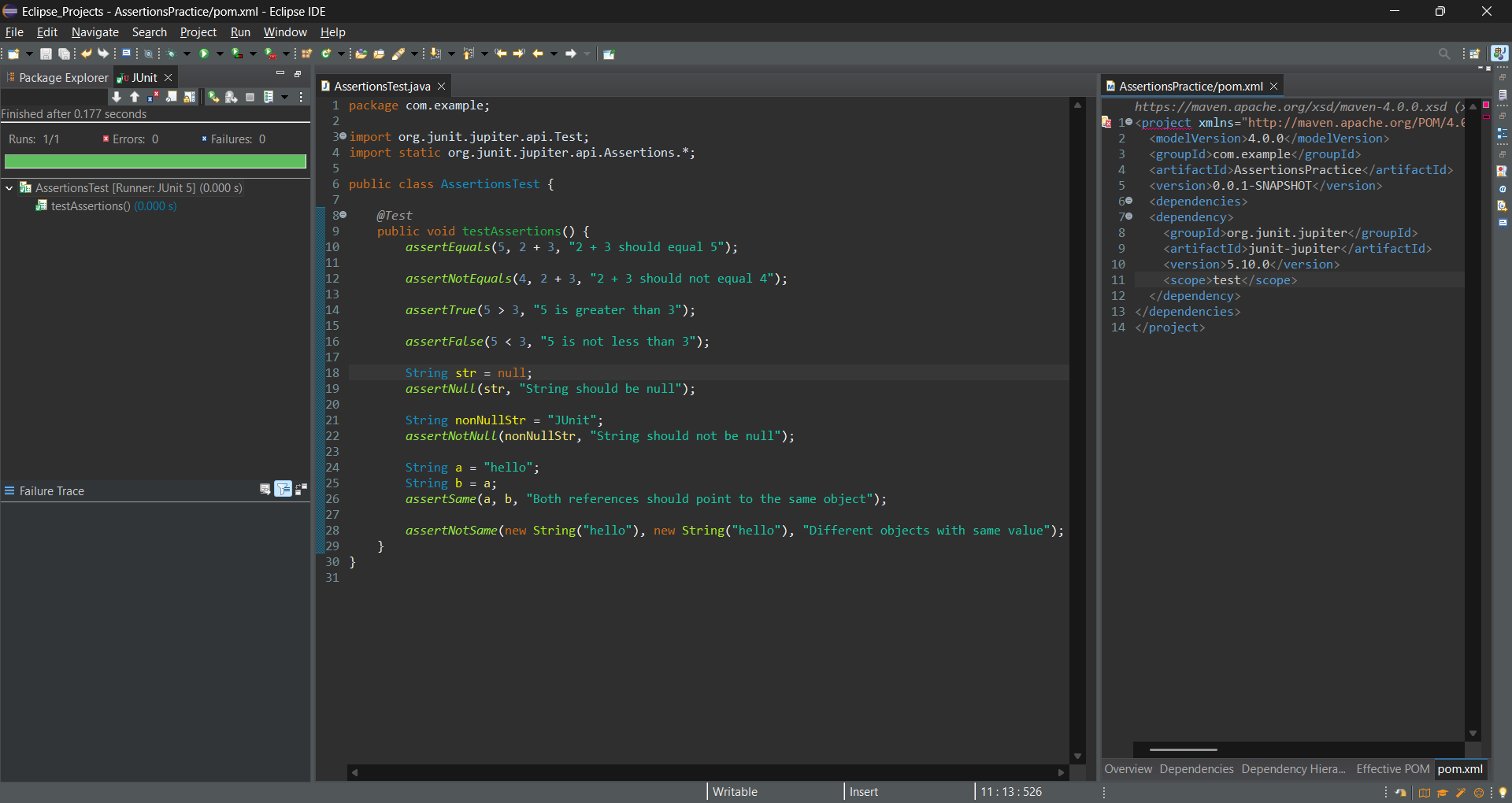
*assertSame*(a, b, "Both references should point to the same object");

*assertNotSame*(new String("hello"), new String("hello"), "Different objects with same value");

}

}

Output:



**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.

**Steps:**

1. Write tests using the AAA pattern.

2. Use @Before and @After annotations for setup and teardown methods.

Program:

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;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) {

throw new ArithmeticException("Division by zero is not allowed.");

}

return a / b;

}

}

CalculatorTest.java

package com.example;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator();

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

}

*@After*

public void tearDown() {

calculator = null;

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

}

*@Test*

public void testAdd() {

int result = calculator.add(2, 3);

*assertEquals*(5, result);

}

*@Test*

public void testSubtract() {

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

*assertEquals*(6, result);

}

}

Output:

