**Exercise 1: Setting Up JUnit**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Exercise1\_SetupTest {

@Test

public void testAdd\_PositiveNumbers() {

// Arrange

Calculator calc = new Calculator();

int a = 2;

int b = 3;

// Act

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

// Assert with custom message

assertEquals("Expected the sum of 2 and 3 to be 5", 5, result);

// Console Output

System.out.println("testAdd\_PositiveNumbers: Passed | " + a + " + " + b + " = " + result);

}

@Test

public void testAdd\_WithZero() {

Calculator calc = new Calculator();

int result = calc.add(0, 7);

assertEquals("Adding zero should return the other number", 7, result);

System.out.println(" testAdd\_WithZero: Passed | 0 + 7 = " + result);

}

@Test

public void testAdd\_NegativeNumbers() {

Calculator calc = new Calculator();

int result = calc.add(-4, -6);

assertEquals("Sum of two negative numbers incorrect", -10, result);

System.out.println("testAdd\_NegativeNumbers: Passed | -4 + -6 = " + result);

}

}

**Calculator.Java**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>exercise1-junit-setup</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

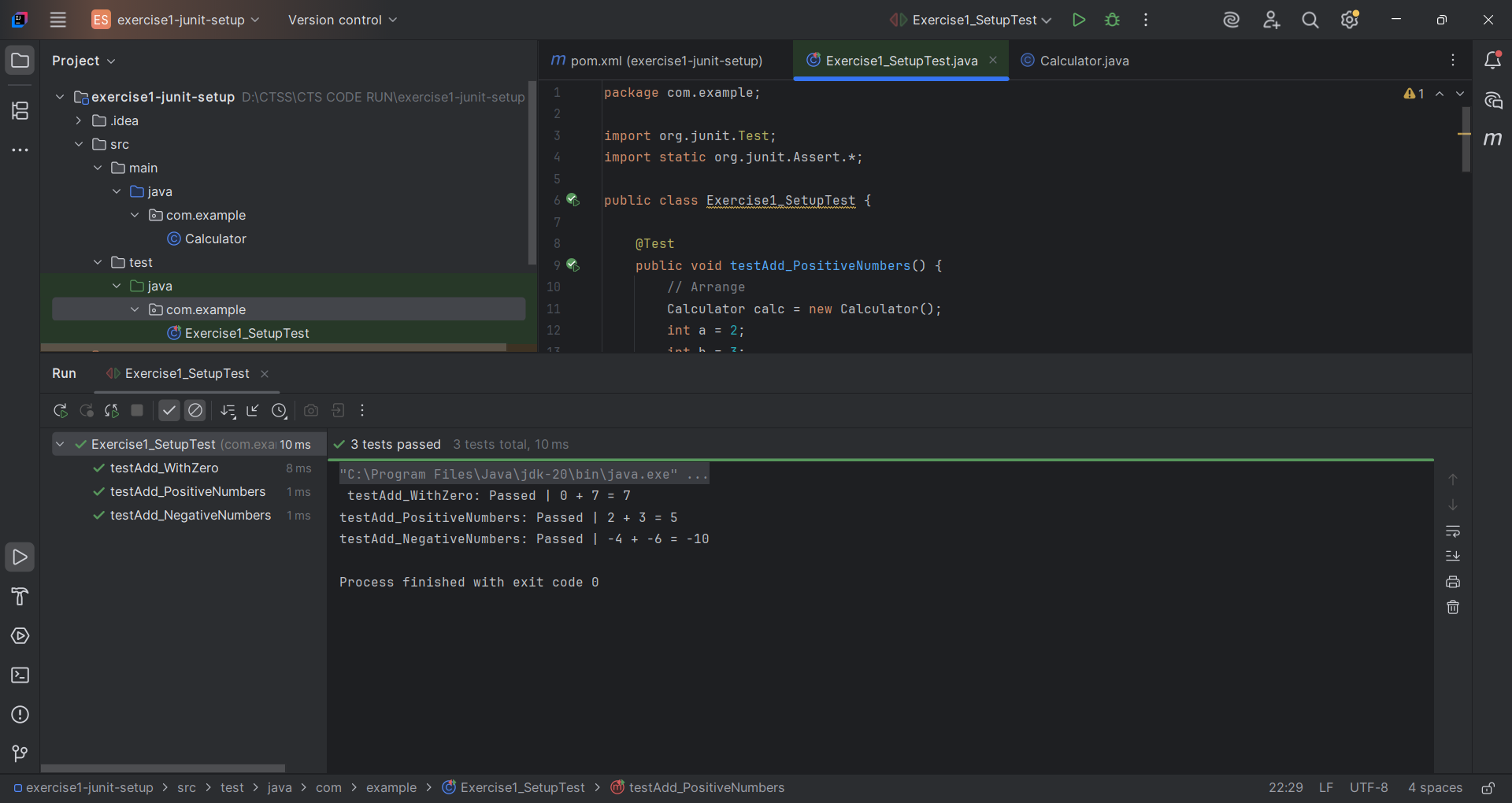
<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**Output :**

**Exercise 3: Assertions in JUnit**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert equals

assertEquals("Sum check failed", 5, 2 + 3);

System.out.println(" Math check: 2 + 3 = 5 confirmed!");

// Assert true

assertTrue("True condition failed", 5 > 3);

System.out.println(" Logical truth: 5 is indeed greater than 3.");

// Assert false

assertFalse("False condition failed", 5 < 3);

System.out.println(" Confirmed: 5 is not less than 3.");

// Assert null

Object nullObj = null;

assertNull("Object should be null", nullObj);

System.out.println(" Memory check: Object is correctly null.");

// Assert not null

Object obj = new Object();

assertNotNull("Object should not be null", obj);

System.out.println(" Memory check: Object is not null and exists.");

System.out.println(" All JUnit assertions executed successfully without failures!");

}

}

**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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>exercise3-assertions-junit</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

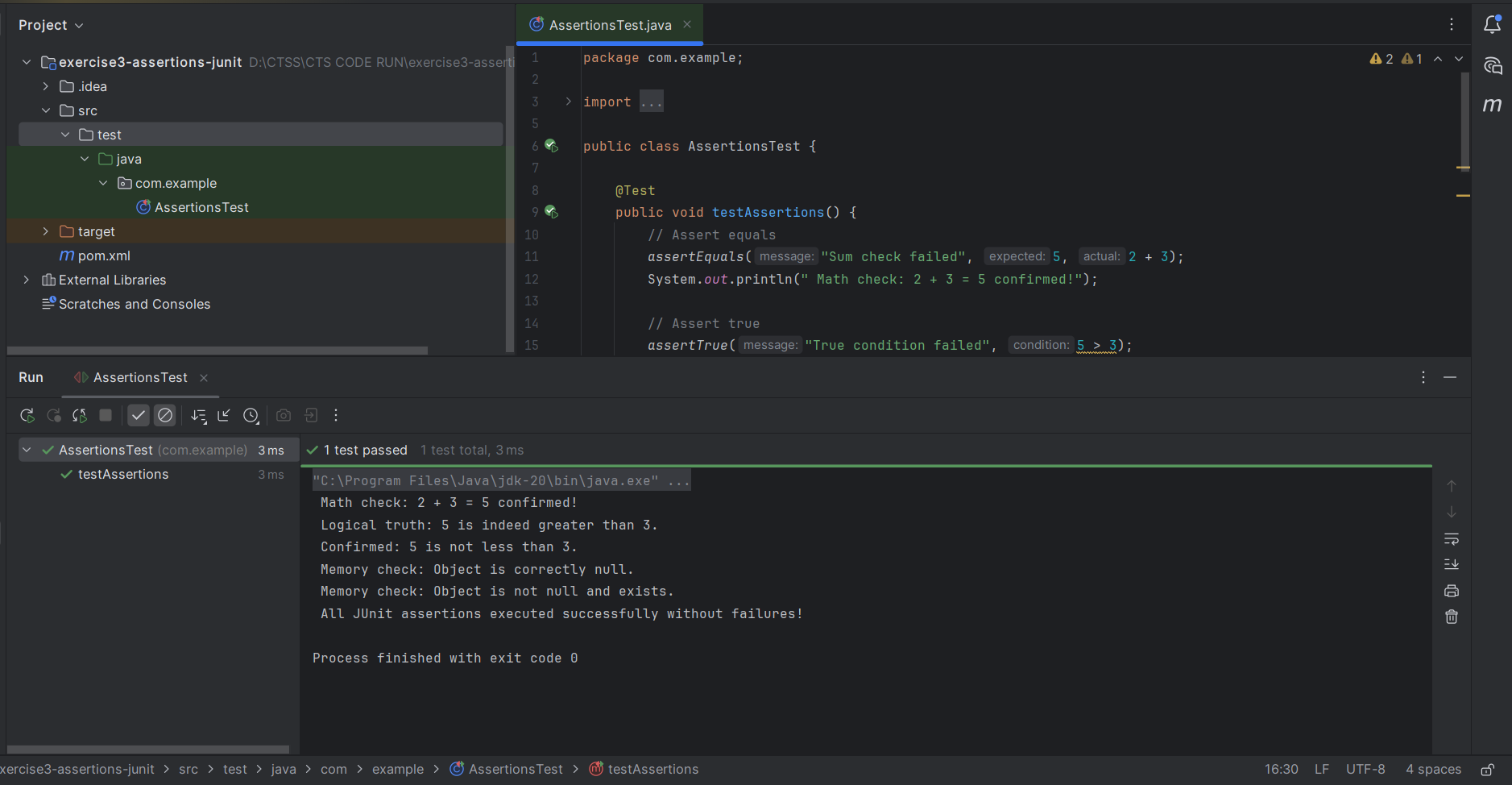
<scope>test</scope>

</dependency>

</dependencies>

</project>

**Output :**

****

**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

package com.example;

import org.junit.After;

import org.junit.Before;

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 complete: Calculator instance created.");

}

@After

public void tearDown() {

System.out.println(" Teardown complete: Calculator instance discarded.\n");

}

@Test

public void testAddition() {

System.out.println(" Arrange: Preparing inputs for addition.");

int a = 10;

int b = 5;

System.out.println(" Act: Performing addition operation.");

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

System.out.println(" Assert: Verifying the result.");

assertEquals("Addition failed", 15, result);

System.out.println(" testAddition passed: " + a + " + " + b + " = " + result);

}

@Test

public void testSubtraction() {

System.out.println(" Arrange: Preparing inputs for subtraction.");

int a = 9;

int b = 4;

System.out.println(" Act: Performing subtraction operation.");

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

System.out.println(" Assert: Verifying the result.");

assertEquals("Subtraction failed", 5, result);

System.out.println(" testSubtraction passed: " + a + " - " + b + " = " + result);

}

@Test

public void testMultiplication() {

System.out.println(" Arrange: Preparing inputs for multiplication.");

int a = 3;

int b = 7;

System.out.println(" Act: Performing multiplication operation.");

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

System.out.println(" Assert: Verifying the result.");

assertEquals("Multiplication failed", 21, result);

System.out.println(" testMultiplication passed: " + a + " \* " + b + " = " + result);

}

}

**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

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>exercise4-aaa-setup</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</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;

}

}

**Output :**

