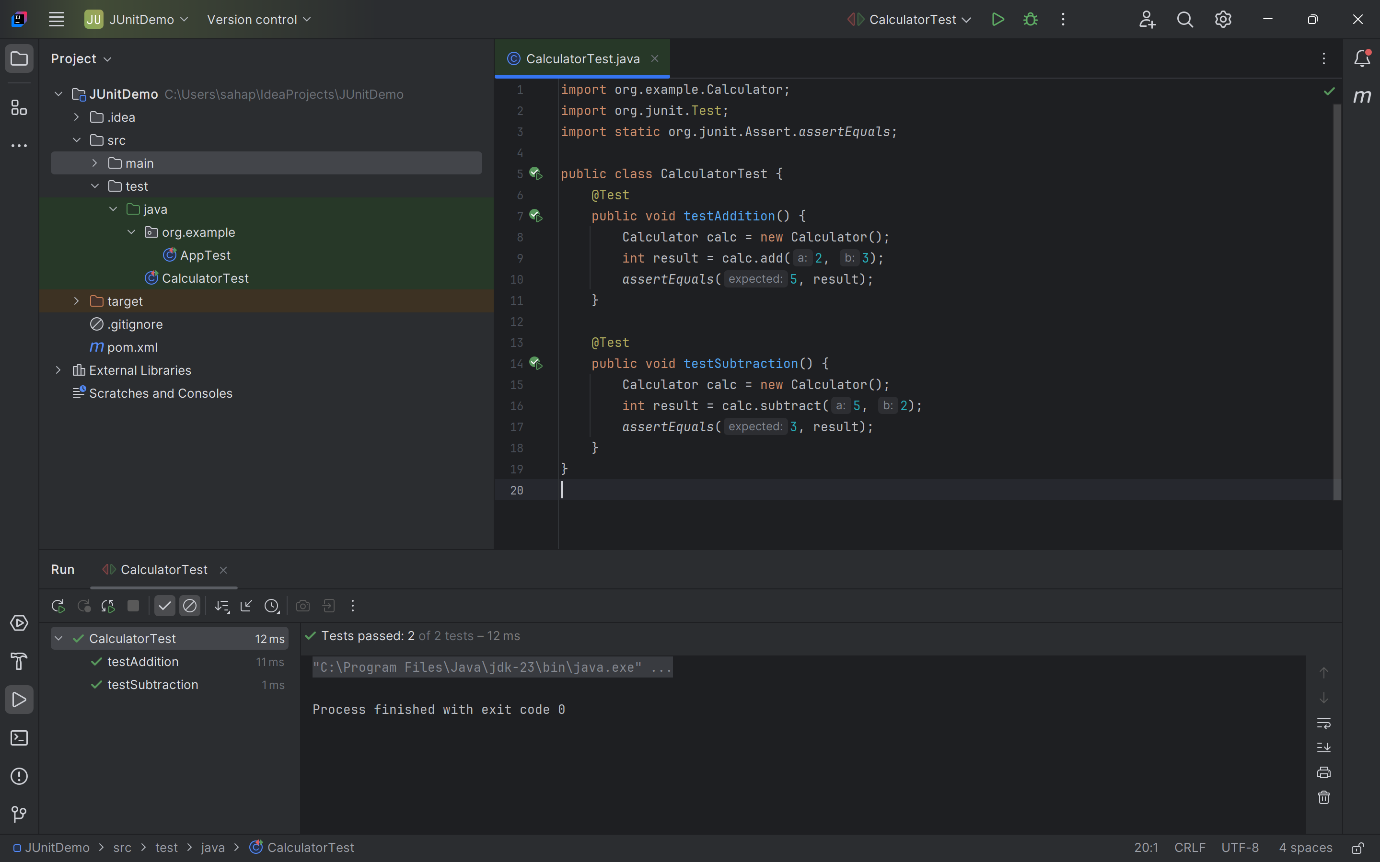
**EXERCISE 1: SETTING UP JUNIT**

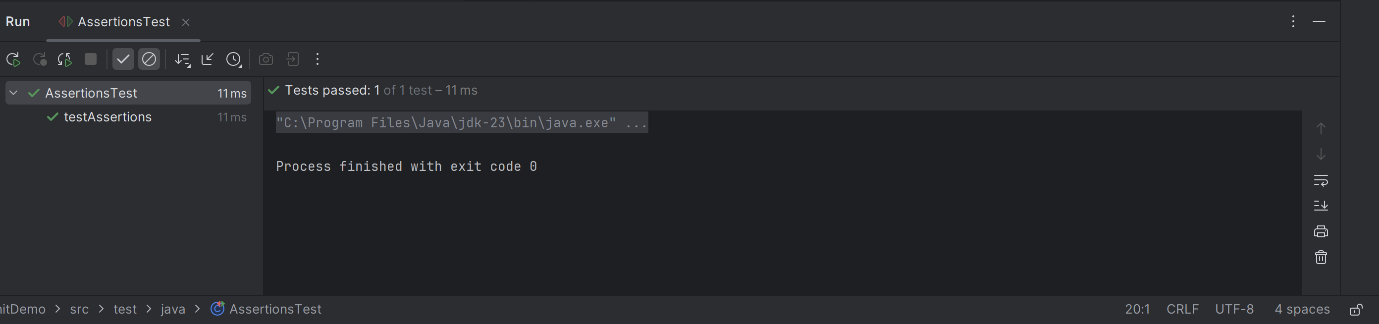
****

**EXERCISE 3: ASSERTIONS IN JUNIT**

**Code:**

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
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());  
 }  
}

**OUTPUT:**

****

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

**Code :**

**Calculator.java –**

package org.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("Cannot divide by zero");  
 return a / b;  
 }  
}

**CalculatorTest.java-**

import org.example.Calculator;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.*assertEquals*;  
  
public class CalculatorTest {  
 private Calculator calculator;  
  
 // Setup method - runs before each test  
 @Before  
 public void setUp() {  
 calculator = new Calculator(); // Arrange  
 System.*out*.println("Setting up Calculator...");  
 }  
  
 // Teardown method - runs after each test  
 @After  
 public void tearDown() {  
 calculator = null;  
 System.*out*.println("Tearing down Calculator...");  
 }  
  
 @Test  
 public void testAddition() {  
  
 // Act  
 int result = calculator.add(3, 4);  
  
 *assertEquals*(7, result);  
 }  
  
 @Test  
 public void testSubtraction() {  
 int result = calculator.subtract(10, 4);  
 *assertEquals*(6, result);  
 }  
  
 @Test  
 public void testMultiplication() {  
 int result = calculator.multiply(2, 5);  
 *assertEquals*(10, result);  
 }  
  
 @Test  
 public void testDivision() {  
 int result = calculator.divide(10, 2);  
 *assertEquals*(5, result);  
 }  
  
 @Test(expected = ArithmeticException.class)  
 public void testDivisionByZero() {  
 calculator.divide(10, 0); // Should throw exception  
 }  
}

**OUTPUT:**

