**WEEK 2**

**JUnit Testing Exercises**

**Exercise 1: Setting Up JUnit**

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>demo</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>

Calculator.java (main class inside- src/main/java/com/example)

package com.example;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

}

CalculatorTest.java (JUnit test class inside- src/test/java/com/example)

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

    @Test

    public void testAdd() {

        Calculator calc = new Calculator();

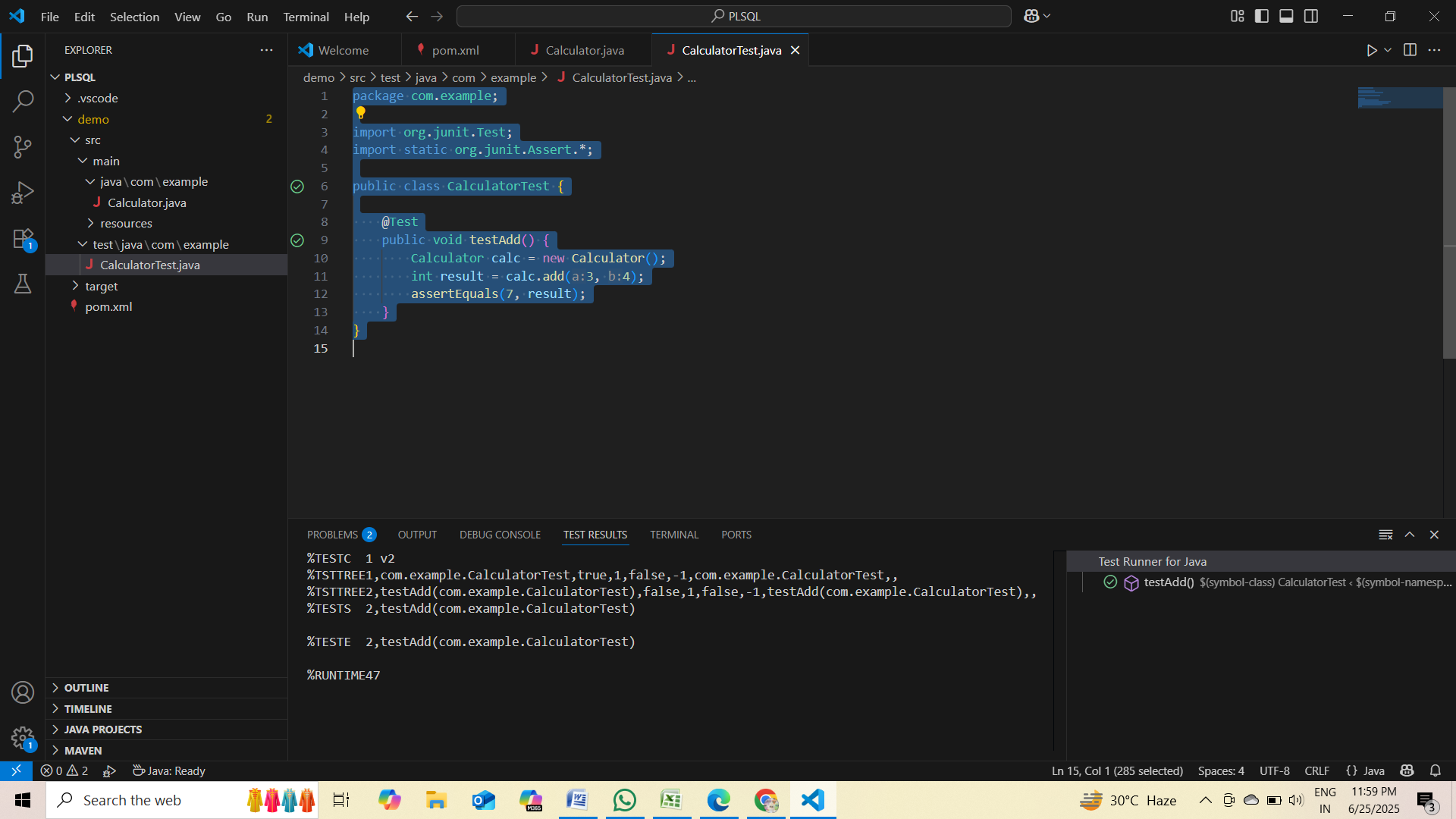
        int result = calc.add(3, 4);

        assertEquals(7, result);

    }

}

**OUTPUT**

****

**Exercise 3: Assertions in JUnit**

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>demo</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>

AssertionsTest.java ( inside- src/test/java/com/example)

package com.example;

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);

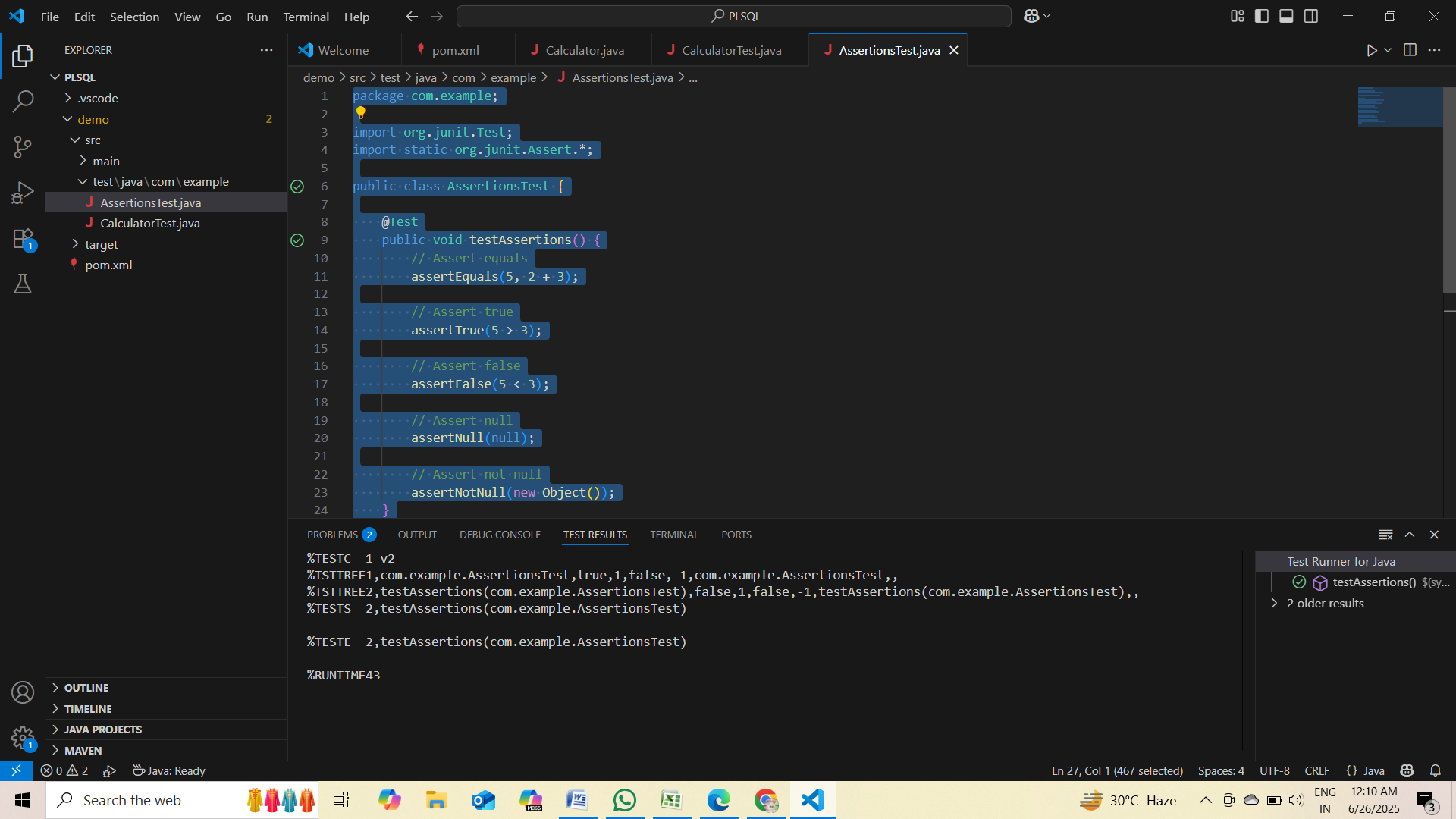
        // Assert not null

        assertNotNull(new Object());

    }

}

**OUTPUT**



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

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>demo</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>

Calculator.java (main class inside- src/main/java/com/example)

package com.example;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

}

CalculatorTestFixture.java (inside- src/test/java/com/example CalculatorTestFixture.java)

package com.example;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTestFixture {

    private Calculator calc;

    // Setup method: runs before each test

    @Before

    public void setUp() {

        calc = new Calculator(); // Arrange

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

    }

    // Teardown method: runs after each test

    @After

    public void tearDown() {

        calc = null;

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

    }

    @Test

    public void testAddition() {

        // Arrange done in setUp()

        // Act

        int result = calc.add(2, 3);

        // Assert

        assertEquals(5, result);

    }

    @Test

    public void testAdditionWithNegatives() {

        // Act

        int result = calc.add(-1, -4);

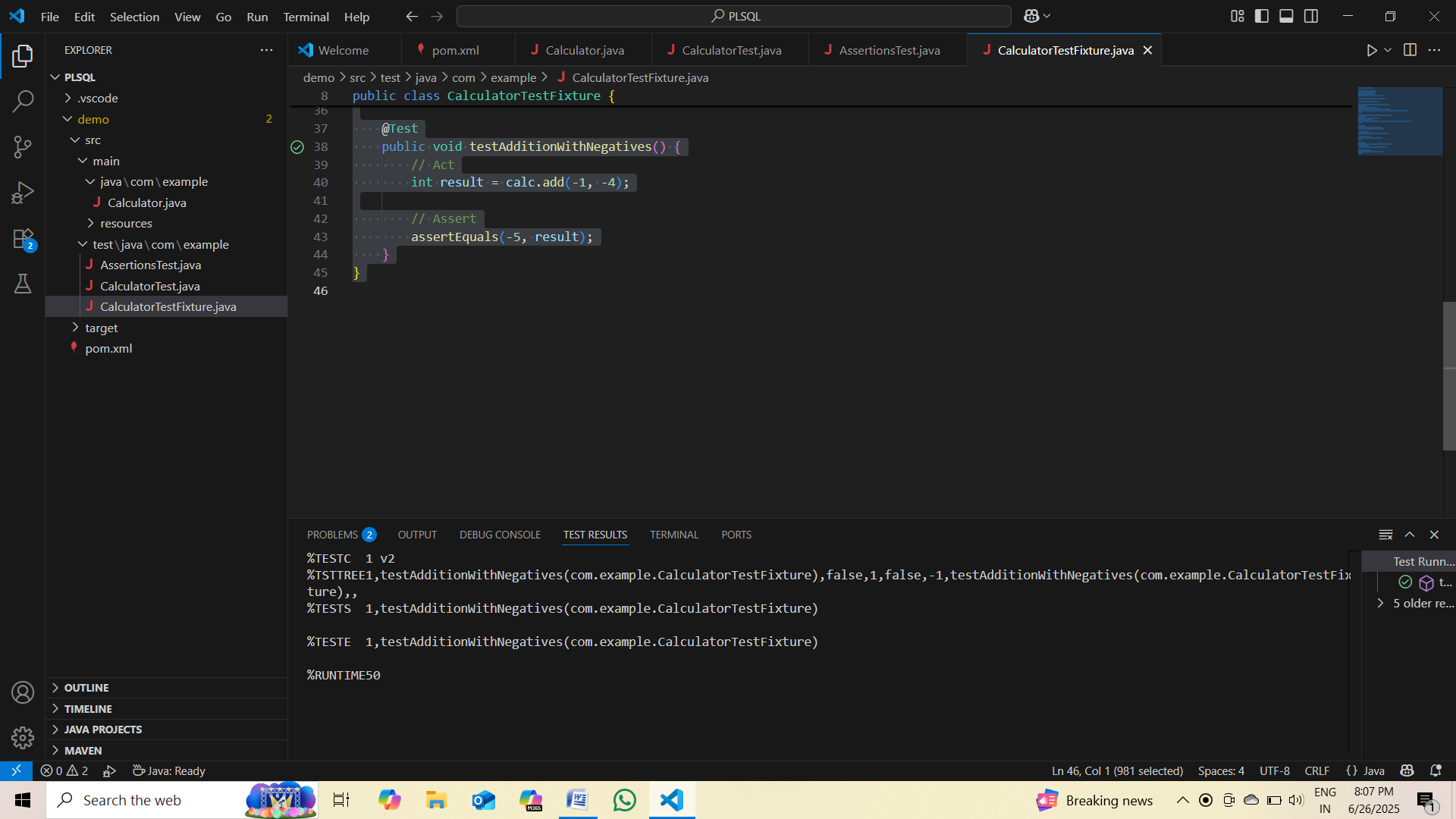
        // Assert

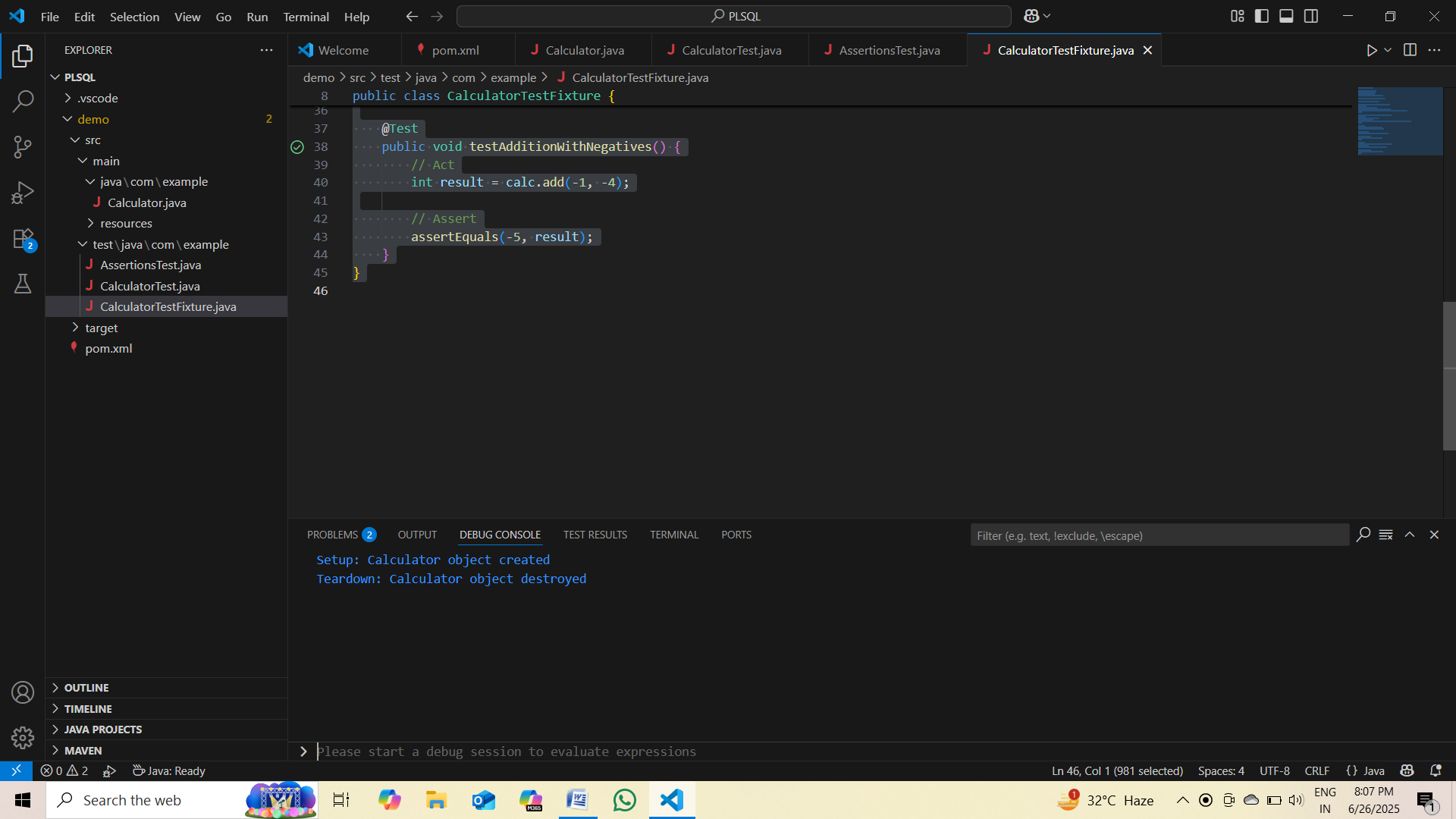
        assertEquals(-5, result);

    }

}

**OUTPUT**

****

****