**eeExercise 1: Setting Up JUnit**

**Step 1:**

**GO TO ECLIPSE**

Launch Eclipse

Select a workspace folder

**Step 2:**

**CREATE A PROJECT**

1. Go to File → New → Other...
2. Type: Maven Project → Click Next
3. Check *Create a simple project (skip archetype selection)* → Next
4. Fill in the following:

| Field | Value |
| --- | --- |
| Group ID | com.example |
| Artifact ID | junitdemo |
| Version | 1.0-SNAPSHOT |

1. Click finish

**Step 3:**

**Add dependencies (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>junitdemo</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>

**Step 4:**

C**REATE JAVA PROGRAM**

Right-click src/main/java → New → Package: com.example

New → Class → Name: Calculator

**Calculator.java**

package com.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**Step 5:**

**CREATE TEST CASES**

Right-click src/test/java → New → Package: com.example

New → JUnit Test Case → Name: CalculatorTest

Select JUnit 4

**CalculatorTest.java**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

@Test

public void testAdd() {

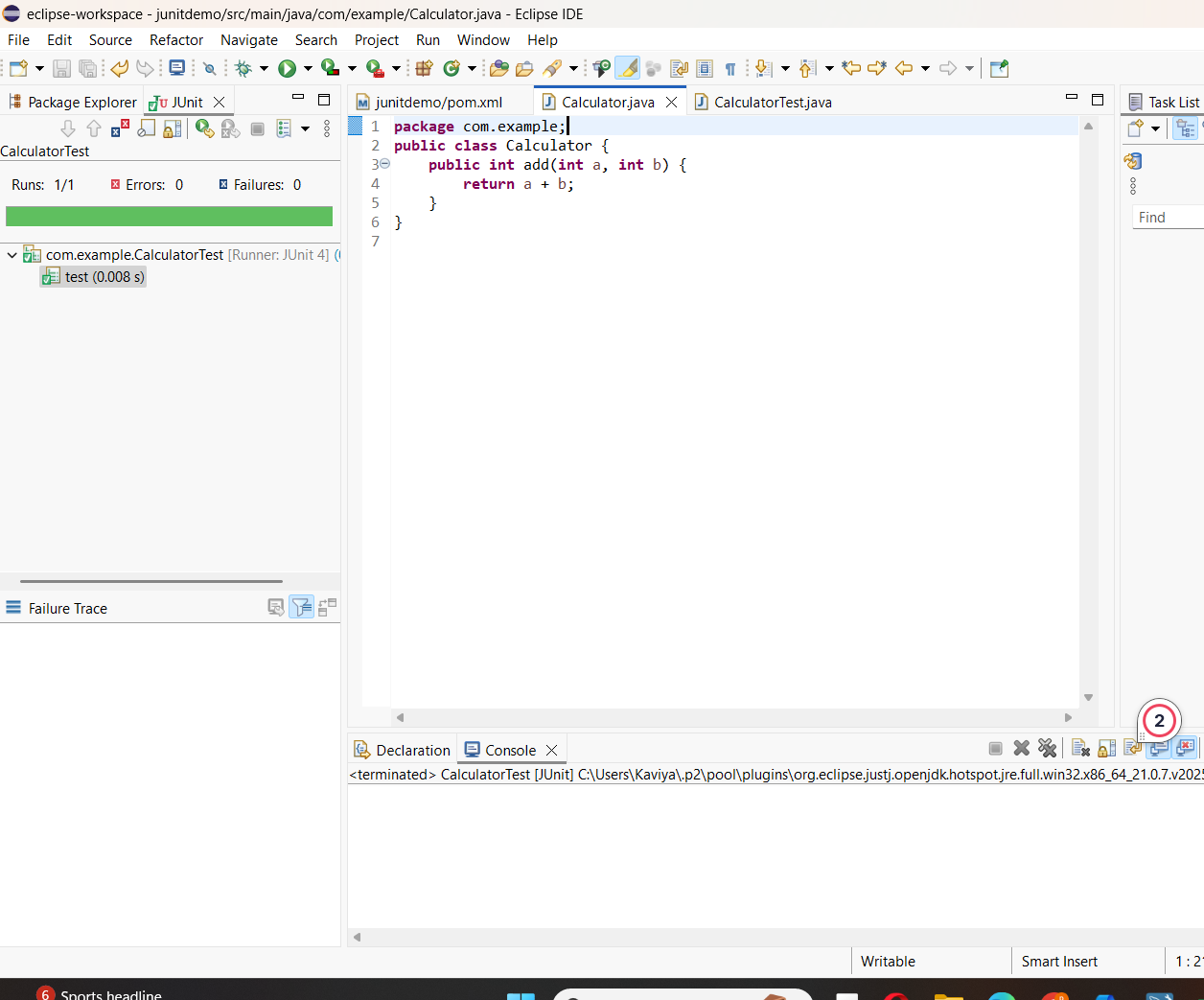
Calculator calc = new Calculator();

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

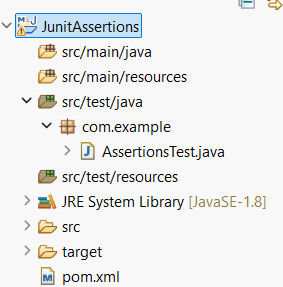
}

}

**OUTPUT**

****

**Exercise 3: Assertions in JUnit**

****

**AssertionsTest.java**

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());

}

}

**Pom.xml**

<dependencies>

<!-- JUnit 4 Testing Library -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- Optional: Needed for assertThat() with Matchers -->

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

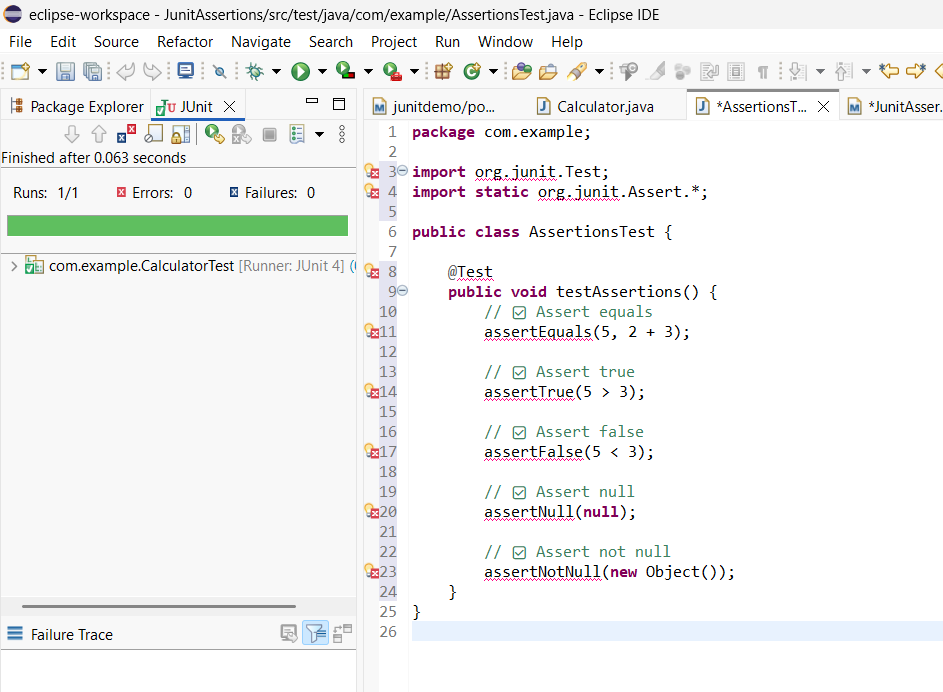
<version>2.2</version>

<scope>test</scope>

</dependency>

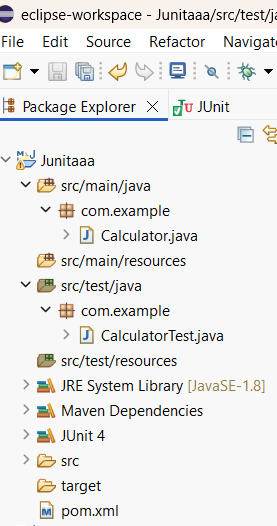
</dependencies>

**OUTPUT:**

****

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

**Teardown Methods in JUnit**

****

**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;

}

}

**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;

// Runs before each test

@Before

public void setUp() {

System.out.println("Setting up...");

calculator = new Calculator();

}

// Runs after each test

@After

public void tearDown() {

System.out.println("Tearing down...");

calculator = null;

}

// Test 1: Addition

@Test

public void testAddition() {

// Arrange

int a = 4;

int b = 6;

// Act

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

// Assert

assertEquals("4 + 6 should equal 10", 10, result);

}

// Test 2: Subtraction

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 3;

// Act

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

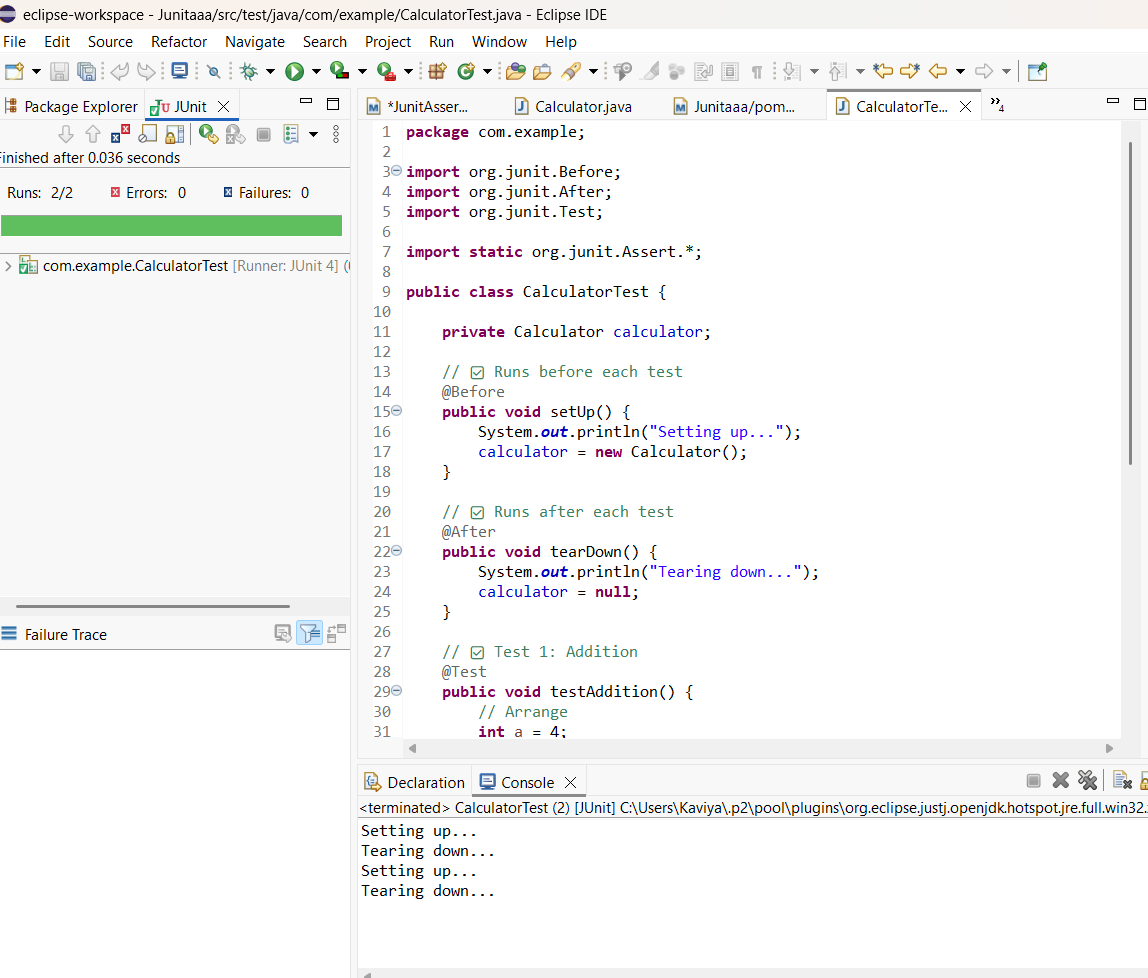
// Assert

assertEquals("10 - 3 should equal 7", 7, result);

}

}

**OUTPUT:**

****