**⇒ Mandatory Hands-On**

**Exercise 1: Setting Up JUnit**

JUnitPractice /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.mycompany</groupId>

<artifactId>JUnitPractice</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>

JUnitPractice.java

package com.mycompany.tests;

import static org.junit.Assert.*assertEquals*;

import org.junit.Test;

public class MyFirstTest {

*@Test*

public void testAddition() {

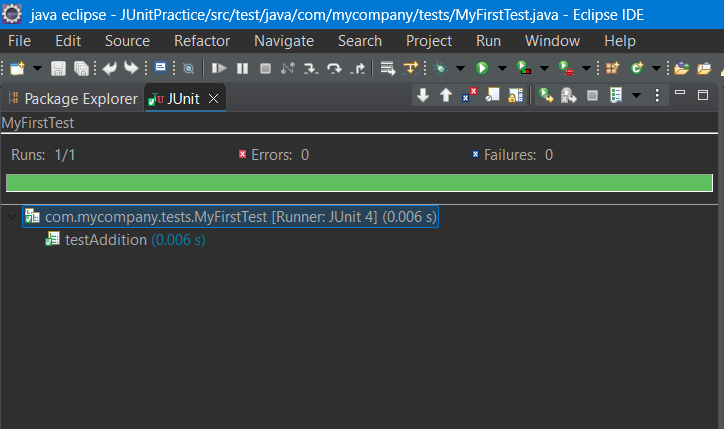
int sum = 2 + 3;

*assertEquals*(5, sum);

}

}

Result



**Exercise 3: Assertions in JUnit**

AssertionsTest.java

package com.mycompany.tests;

import static org.junit.Assert.\*;

import org.junit.Test;

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);

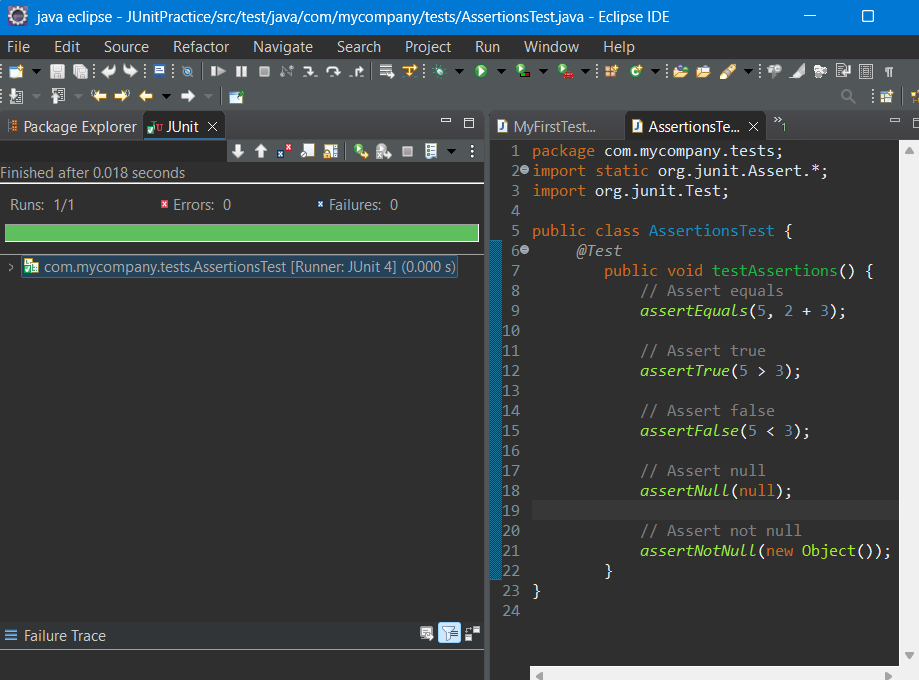
// Assert not null

*assertNotNull*(new Object());

}

}

Result



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

Calculator.java

package com.mycompany.tests;

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.mycompany.tests;

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import com.mycompany.\*;

public class CalculatorTest {

private Calculator calculator;

*@Before*

public void setUp() {

// Arrange — setup before each test

calculator = new Calculator();

System.***out***.println("Setup done");

}

*@After*

public void tearDown() {

// Teardown — cleanup after each test

calculator = null;

System.***out***.println("Teardown done");

}

*@Test*

public void testAddition() {

// Arrange is done in setUp()

// Act

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

// Assert

*assertEquals*(5, result);

}

*@Test*

public void testSubtraction() {

// Arrange is done in setUp()

// Act

int result = calculator.subtract(5, 3);

// Assert

*assertEquals*(2, result);

}

}

Result

