**JUnit Testing Exercises**

**Exercise 1: Setting Up JUnit**

**Scenario:** You need to set up JUnit in your Java project to start writing unit tests.

**JUnit Test Class**

package com.test;

import org.junit.Test;

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

public class CalculatorTest {

@Test

public void testAddition() {

int a = 2 + 3;

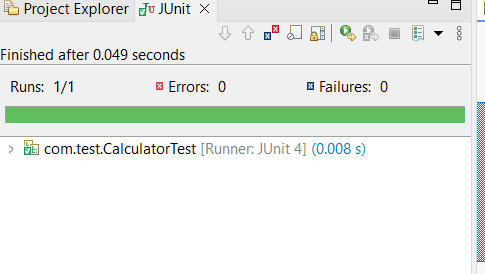
System.*out*.println("The result is: " + a);

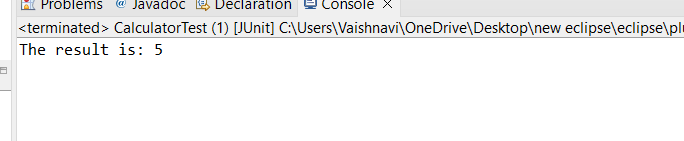
*assertEquals*(5, a);

}

}

**Output:**

****

****

**NOTE:**

Configured JUnit without Maven by manually adding JAR files, writing a test class, and executing it using Eclipse. The test passed successfully and the output was printed in the console**.**

**Exercise 3: Assertions in Junit**

**Scenario:** You need to use different assertions in JUnit to validate your test results.

**Assertion Types Using Junit**

package com.test;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

// assertEquals

int sum = 2 + 3;

System.*out*.println("Sum: " + sum);

*assertEquals*(5, sum);

// assertTrue

boolean conditionTrue = (10 > 5);

System.*out*.println("Is 10 > 5? " + conditionTrue);

*assertTrue*(conditionTrue);

// assertFalse

boolean conditionFalse = (5 < 3);

System.*out*.println("Is 5 < 3? " + conditionFalse);

*assertFalse*(conditionFalse);

// assertNull

String nullString = null;

System.*out*.println("Null string: " + nullString);

*assertNull*(nullString);

// assertNotNull

String notNullString = "Vaishu";

System.*out*.println("Not null string: " + notNullString);

*assertNotNull*(notNullString);

// assertSame

String str1 = "OpenAI";

String str2 = str1;

System.*out*.println("str1 and str2 refer to the same object");

*assertSame*(str1, str2);

// assertNotSame

String str3 = new String("Test");

String str4 = new String("Test");

System.*out*.println("str3 and str4 do not refer to the same object");

*assertNotSame*(str3, str4);

// assertArrayEquals

int[] expectedArray = {1, 2, 3};

int[] actualArray = {1, 2, 3};

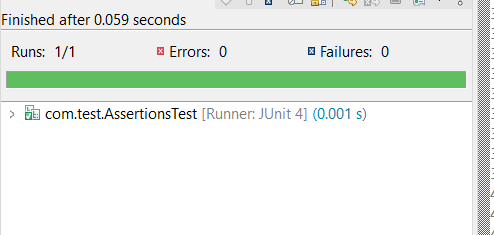
System.*out*.println("Comparing two integer arrays");

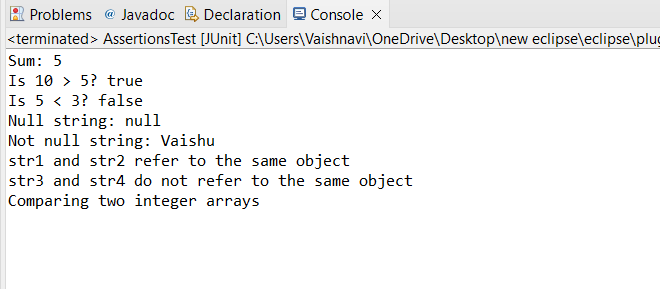
*assertArrayEquals*(expectedArray, actualArray);

}

}

**OUTPUT:**

****

****

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

**Scenario:**You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

**@Before and @After annotations for setup and teardown methods.**

package com.test;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTesting {

private int a;

private int b;

// Setup method – runs before every test

@Before

public void setUp() {

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

a = 10;

b = 5;

}

// Teardown method – runs after every test

@After

public void tearDown() {

System.*out*.println("Cleaning up after test...");

a = 0;

b = 0;

}

// Test addition using AAA pattern

@Test

public void testAddition() {

// Arrange – already done in @Before

// Act

int result = a + b;

System.*out*.println("Addition Result: " + result);

// Assert

*assertEquals*(15, result);

}

// Test subtraction using AAA pattern

@Test

public void testSubtraction() {

// Act

int result = a - b;

System.*out*.println("Subtraction Result: " + result);

// Assert

*assertEquals*(5, result);

}

}

**OUTPUT:**

