**JUnit Testing**

**Exercise – 1: Setting Up JUnit**

//pom.xml

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>MyJUnitProject</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>

//CalculatorTest.java

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class CalculatorTest {

@Test

public void testAddition() {

int a = 2;

int b = 3;

int sum = a + b;

assertEquals(5, sum);

}

}

**Output:**

----------------------------------

JUnit

----------------------------------

testAddition()

✔ passed

----------------------------------

Runs: 1/1 Errors: 0 Failures: 0

**Exercise – 2: Assertions in Junit**

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert equals

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

assertFalse(5 < 3);

assertNull(null);

assertNotNull(new Object());

}

}

Output:

-------------------------------

JUnit

-------------------------------

testAssertions()

✔ passed

-------------------------------

Runs: 1/1 Errors: 0 Failures: 0

**Exercise – 3: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit**

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

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

calculator = new Calculator();

}

@After

public void tearDown() {

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

calculator = null;

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

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

assertEquals(8, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

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

assertEquals(6, result);

}

}

// Simple Calculator class for demonstration

class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**Output:**

Setting up Calculator instance...

Cleaning up after test...

Setting up Calculator instance...

Cleaning up after test...

BUILD SUCCESSFUL in 1s

2 tests completed, 0 failed