JUnit\_Basic Testing Exercises : Exercise 1: Setting Up Junit

**pom.xml file**

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

**Create the following Java test class to verify setup:**

import org.junit.Test;

import static org.junit.Assert.\*;

public class HelloWorldTest {

@Test

public void testHello() {

String message = "Hello JUnit!";

assertEquals("Hello JUnit!", message);

}

}

**Output :**

Running HelloWorldTest...

testHello() passed

BUILD SUCCESSFUL

Total tests run: 1, Failures: 0, Skipped: 0

JUnit\_Basic Testing Exercises : Exercise 3: 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);

// Assert true

assertTrue(5 > 3);

// Assert false

assertFalse(5 < 3);

// Assert null

assertNull(null);

// Assert not null

assertNotNull(new Object());

}

}

**Output:**

Running AssertionsTest...

testAssertions() passed all assertions:

- assertEquals: 5 == 2 + 3

- assertTrue: 5 > 3

- assertFalse: 5 < 3

- assertNull: null

- assertNotNull: new Object()

BUILD SUCCESSFUL

Total tests run: 1, Failures: 0, Skipped: 0

JUnit\_Basic Testing Exercises : Exercise 4: 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.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup method (runs before each test)

@Before

public void setUp() {

calculator = new Calculator();

System.out.println("Setup complete.");

}

// Teardown method (runs after each test)

@After

public void tearDown() {

System.out.println("Test finished.");

}

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

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

// Assert

assertEquals(8, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

// Act

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

// Assert

assertEquals(6, result);

}

}

**Calculator class :**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**Output :**

Running CalculatorTest...

Setup complete.

testAddition() passed (5 + 3 = 8)

Test finished.

Setup complete.

testSubtraction() passed (10 - 4 = 6)

Test finished.

BUILD SUCCESSFUL

Total tests run: 2, Failures: 0, Skipped: 0