JUnit Testing Exercises

**Exercise 1: Setting Up JUnit**

**Step 1: Create a New Java Project**

* Open IntelliJ IDEA, Eclipse, or any IDE you prefer.
* Create a New Project → Select Maven (recommended for easier dependency management).

## ****Step 2: Add JUnit Dependency (Maven Example)****

In your pom.xml, add:

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

## ****Step 3: Create a Java Class to Test****

Example class:

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

## ****Step 4: Create a Test Class****

In src/test/java, create:

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

@Test

public void testAdd() {

Calculator calc = new Calculator();

int result = calc.add(5, 3);

assertEquals(8, result);

}

}

## ****Step 5: Run the Test****

* Right-click on the test class → Run  
  OR
* Use your IDE's test runner (e.g., green checkmark in IntelliJ).

**Exercise 3: Assertions in JUnit**

Code:

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

@Test

public void testAssertions() {

// Assert that two values are equal

assertEquals(5, 2 + 3);

// Assert that a condition is true

assertTrue(5 > 3);

// Assert that a condition is false

assertFalse(5 < 3);

// Assert that an object is null

String str = null;

assertNull(str);

// Assert that an object is not null

String text = "JUnit";

assertNotNull(text);

}

}

## ****Running the Test****

* Run using your IDE's Test Runner.
* All assertions should pass, and output shows **green** indicating success.

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

Code:

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calc;

// Setup method runs before each test

@Before

public void setUp() {

calc = new Calculator();

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

}

// Teardown method runs after each test

@After

public void tearDown() {

calc = null;

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

}

// Test using Arrange-Act-Assert (AAA) Pattern

@Test

public void testAddition() {

// Arrange

int a = 5;

int b = 3;

// Act

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

// Assert

assertEquals(8, result);

}

@Test

public void testSubtraction() {

// Arrange

int a = 10;

int b = 4;

// Act

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

// Assert

assertEquals(6, result);

}

}

**Supporting 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;

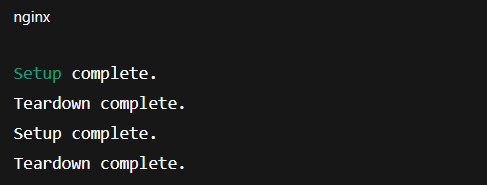
}

}

### AAA Pattern:

* **Arrange**: Setup data or environment
* **Act**: Call the method under test
* **Assert**: Validate the result

**When running tests:**

****