JUnit Testing Exercises

# Exercise 1: Setting Up JUnit

Scenario:

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

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).
2. Add JUnit dependency to your project. If you are using Maven, add the following to your pom.xml:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

1. Create a new test class in your project.

// src/test/java/com/example/CalculatorTest.java

public class CalculatorTest {

// Tests will go here

}

# Exercise 2: Writing Basic JUnit Tests

Scenario:

You need to write basic JUnit tests for a simple Java class. Steps:

1. Create a new Java class with some methods to test.
2. Write JUnit tests for these methods.

# Exercise 3: Assertions in JUnit

Scenario:

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

# 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.

Steps:

1. Write tests using the AAA pattern.
2. Use @Before and @After annotations for setup and teardown methods.

**sample code**

**calculator.java**

package com.example.junit\_demo;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) throw new IllegalArgumentException("Cannot divide by zero");

return a / b;

}

}

**calculatorTest.java**

package com.example.junit\_demo;

import static org.junit.Assert.\*;

import org.junit.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup - runs before each test

*@Before*

public void setUp() {

calculator = new Calculator();

System.***out***.println("Setup for test");

}

// Teardown - runs after each test

*@After*

public void tearDown() {

calculator = null;

System.***out***.println("Teardown after test");

}

// --- Basic Tests with AAA Pattern and Assertions ---

*@Test*

public void testAdd() {

// Arrange done in @Before

// Act

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

// Assert

*assertEquals*(5, result);

}

*@Test*

public void testSubtract() {

*assertEquals*(2, calculator.subtract(5, 3));

}

*@Test*

public void testMultiply() {

*assertEquals*(15, calculator.multiply(3, 5));

}

*@Test*

public void testDivide() {

*assertEquals*(4, calculator.divide(8, 2));

}

*@Test*(expected = ArithmeticException.class)

public void testDivideByZero() {

calculator.divide(10, 0);

}

// --- Additional Assertions Demo ---

*@Test*

public void testAssertionsDemo() {

*assertTrue*(calculator.add(1, 1) == 2);

*assertFalse*(calculator.subtract(2, 2) == 1);

*assertNotNull*(calculator);

*assertNull*(null);

}

}

