# **JUnit Testing Exercises**

**Exercise 1: Setting Up JUnit:**

1. **The below code is added to the pom.xml**

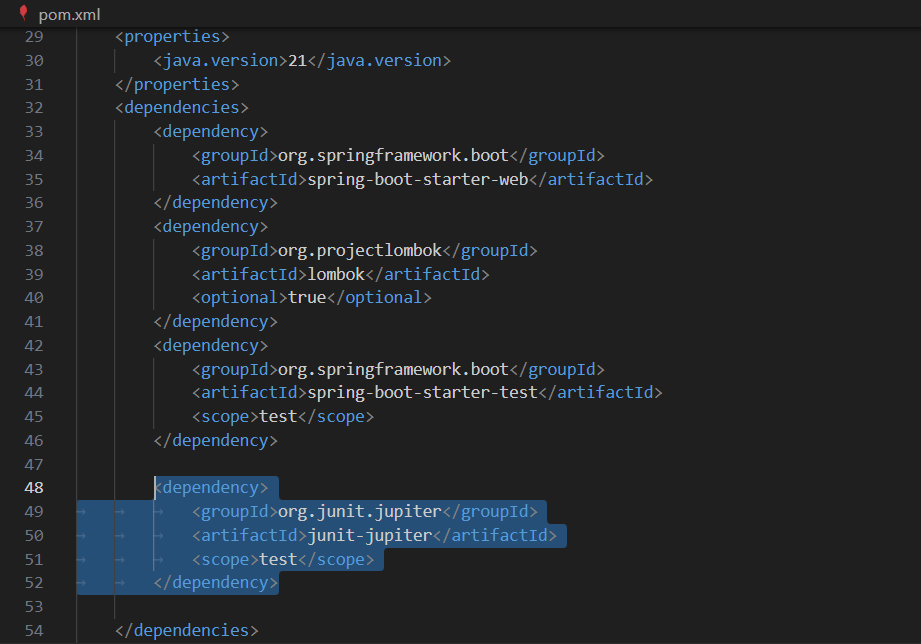
<dependency>

            <groupId>org.junit.jupiter</groupId>

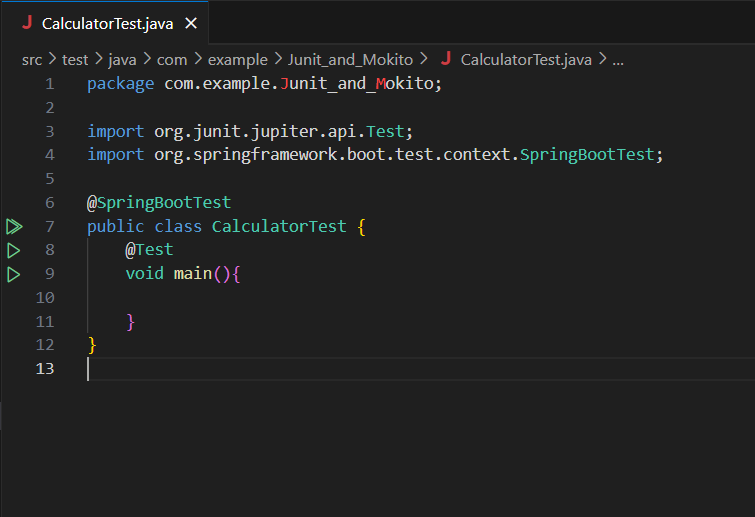
            <artifactId>junit-jupiter</artifactId>

            <scope>test</scope>

</dependency>



1. **Create a test class**

****

**Exercise 3: Assertions in JUnit**

**Code:**

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);

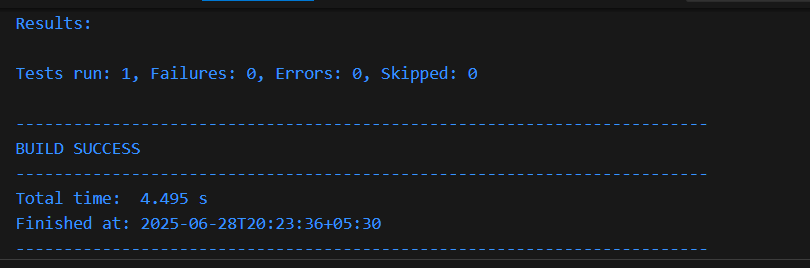
        // Assert not null

        assertNotNull(new Object());

    }

}

**Output:**

****

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

**Code:**

package com.example.Junit\_and\_Mokito;

import org.junit.jupiter.api.AfterEach;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.junit.jupiter.api.Assertions.assertThrows;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

class CalculatorTest {

    Calculator calculator;

    @BeforeEach

    void setUp() {

        calculator = new Calculator(); // Arrange

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

    }

    @AfterEach

    void tearDown() {

        calculator = null;

        System.out.println("Tearing down Calculator instance...\n");

    }

    @Test

    void testAddition() {

        // Arrange: done in setup

        // Act

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

        // Assert

        assertEquals(5, result);

    }

    @Test

    void testSubtraction() {

        int result = calculator.subtract(5, 3);

        assertEquals(2, result);

    }

    @Test

    void testMultiplication() {

        int result = calculator.multiply(4, 3);

        assertEquals(12, result);

    }

    @Test

    void testDivision() {

        int result = calculator.divide(10, 2);

        assertEquals(5, result);

    }

    @Test

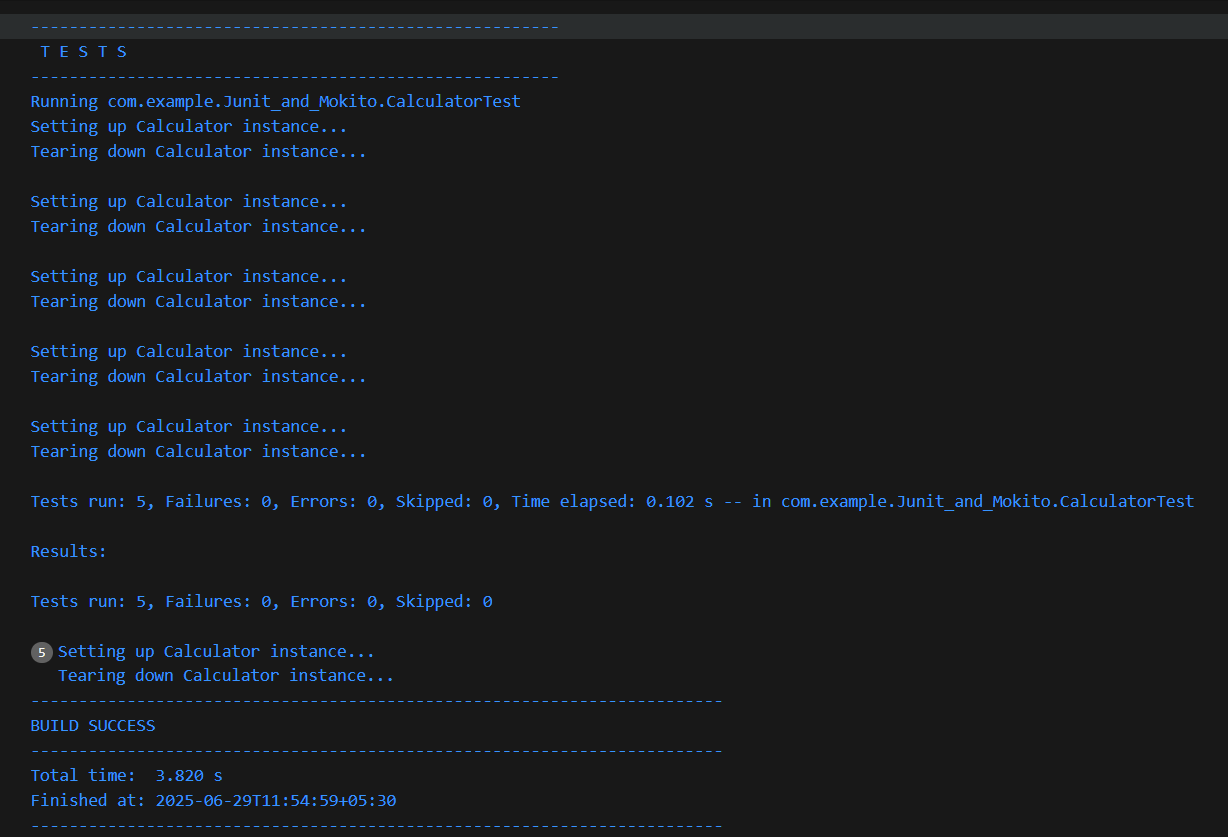
    void testDivisionByZero() {

        assertThrows(ArithmeticException.class, () -> calculator.divide(8, 0));

    }

}

**Output:**

****