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

**🔹 AAA Pattern (Arrange-Act-Assert)**

This pattern breaks the test into 3 parts:

**Arrange** – Set up the data and environment

**Act** – Execute the code being tested

**Assert** – Check that the result is correct

**🔹 @Before and @After**

* @Before → Runs **before each test** (used for setup)
* @After → Runs **after each test** (used for cleanup)

Code

package org.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

// Setup method

@Before

public void setUp() {

calculator = new Calculator();

System.out.println("Setting up before a test...");

}

// Teardown method

@After

public void tearDown() {

calculator = null;

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

}

@Test

public void testAdd() {

// Arrange is already done in @Before

// Act

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

// Assert

assertEquals(5, result);

}

@Test

public void testSubtract() {

int result = calculator.subtract(10, 4);

assertEquals(6, result);

}

}

**Calculator Class (in src/main/java/org/example/Calculator.java)**

package org.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

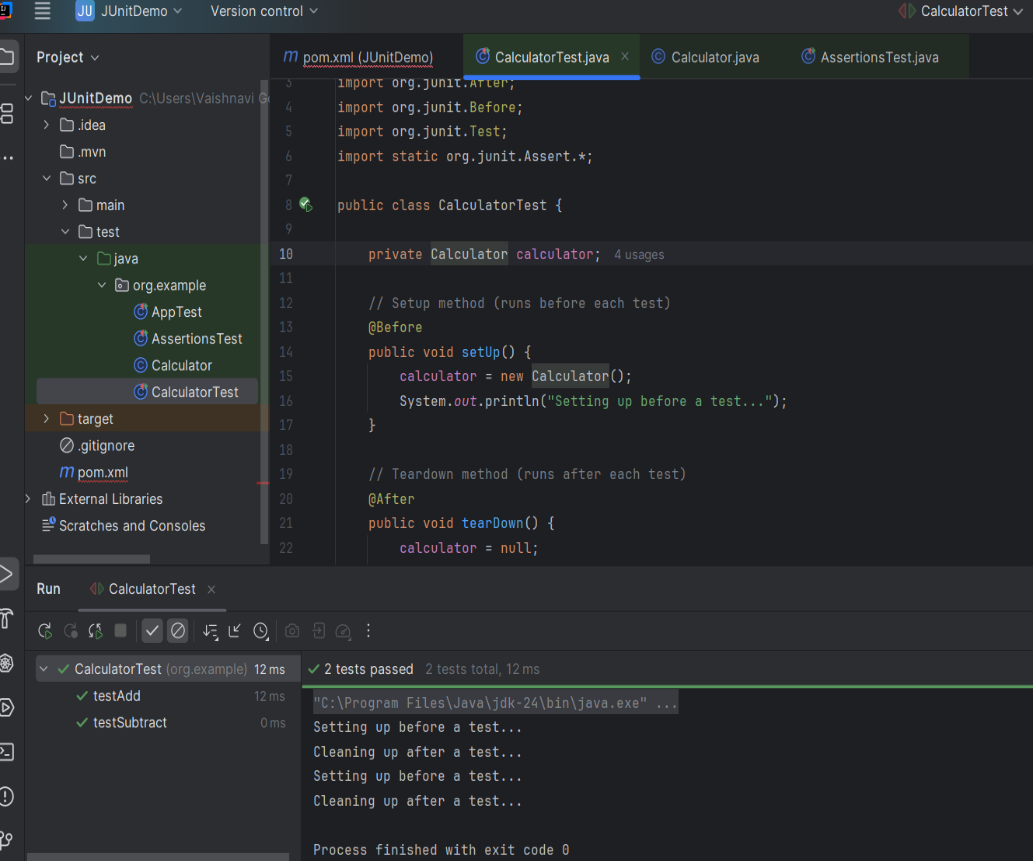
}

}

**Run the Tests**

* Right-click on CalculatorTest → Run 'CalculatorTest'

**OUTPUT**

****