**Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and**

**Teardown Methods in Junit**

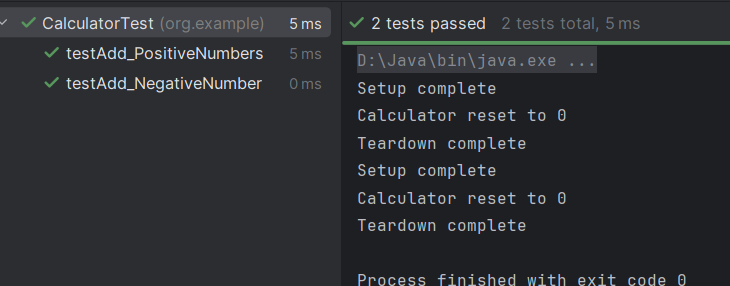
**Calculator.java**

package org.example;  
  
public class Calculator {  
 private int lastResult;  
  
 public int add(int a, int b) {  
 lastResult = a + b;  
 return lastResult;  
 }  
  
 public int getLastResult() {  
 return lastResult;  
 }  
  
 public void reset() {  
 lastResult = 0; // Logic to reset internal state  
 System.*out*.println("Calculator reset to 0");  
 }  
}

**CalculatorTester.java**

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;  
  
 @Before  
 public void setUp() {  
 calculator = new Calculator();  
 System.*out*.println("Setup complete");  
 }  
  
 @After  
 public void tearDown() {  
 calculator.reset();  
 *assertEquals*(0, calculator.getLastResult());   
 System.*out*.println("Teardown complete");  
 }  
  
 @Test  
 public void testAdd\_PositiveNumbers() {  
 int result = calculator.add(2, 3);  
  
 *assertEquals*(5, result);  
 *assertEquals*(5, calculator.getLastResult());   
 }  
  
 @Test  
 public void testAdd\_NegativeNumber() {  
 int result = calculator.add(-1, 4);  
 *assertEquals*(3, result);  
 }  
}

**Output:**

****