**JUnit\_Basic Testing Exercises**

**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

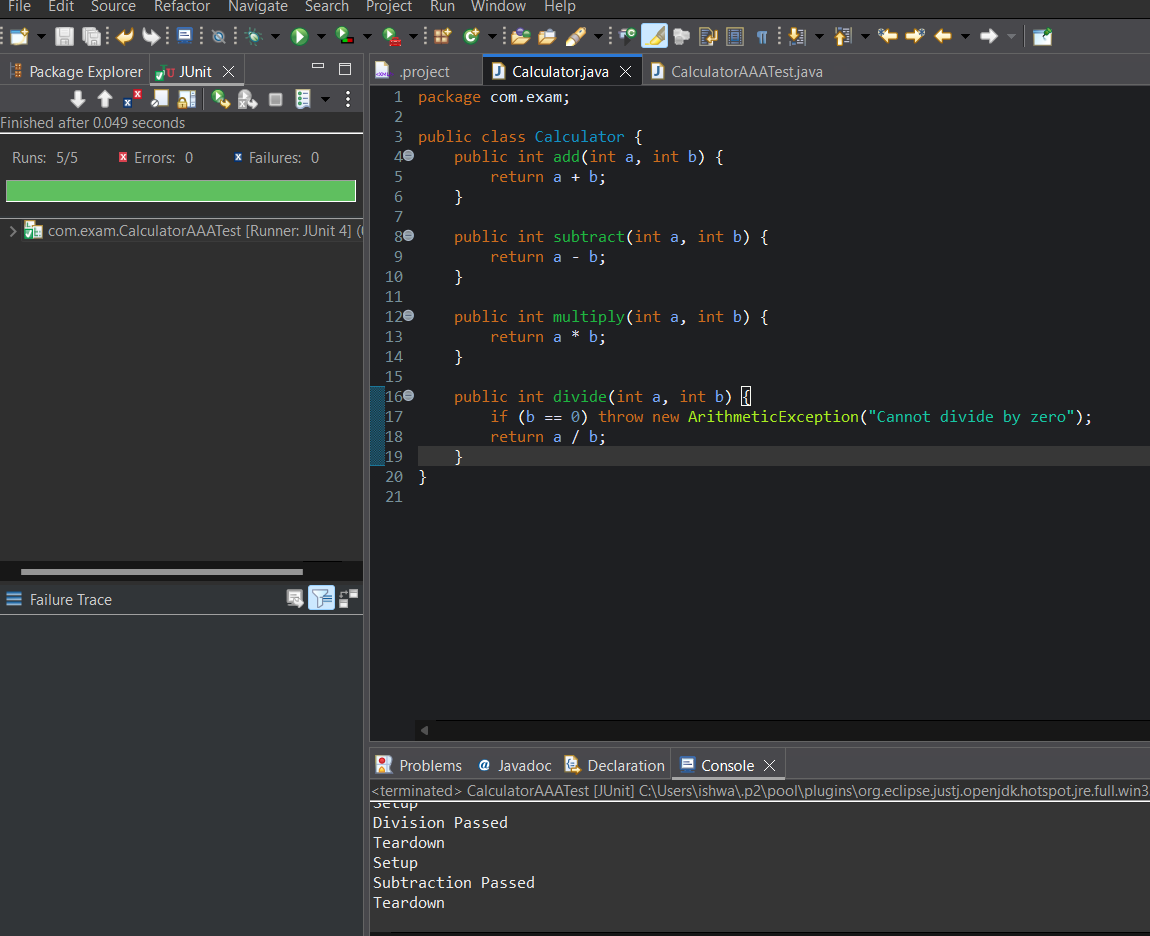
and teardown methods.

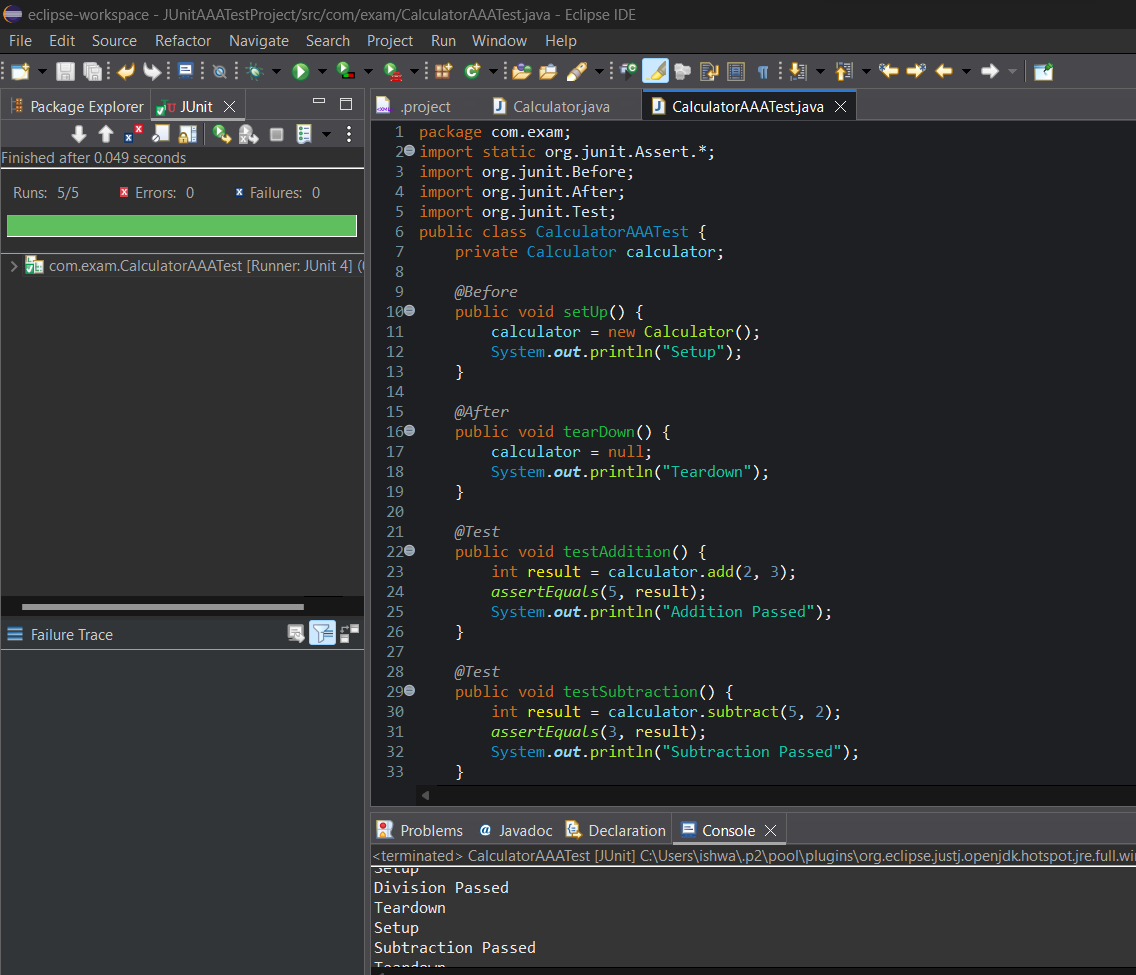
Steps:

1. Write tests using the AAA pattern.

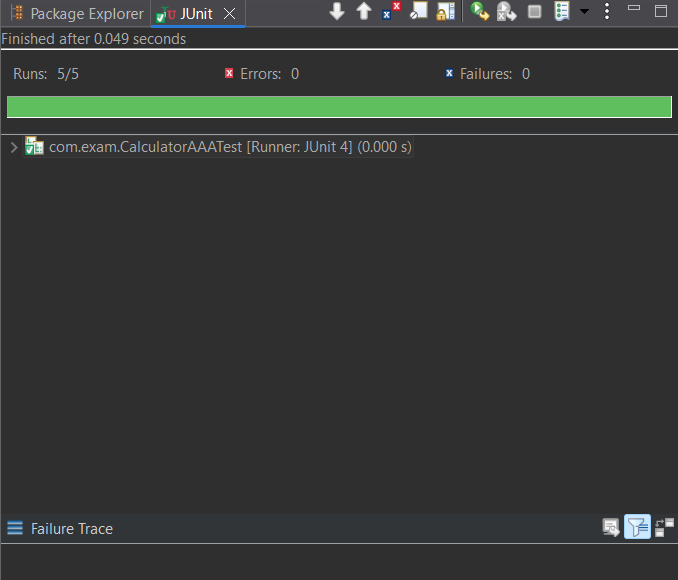
2. Use @Before and @After annotations for setup and teardown methods.

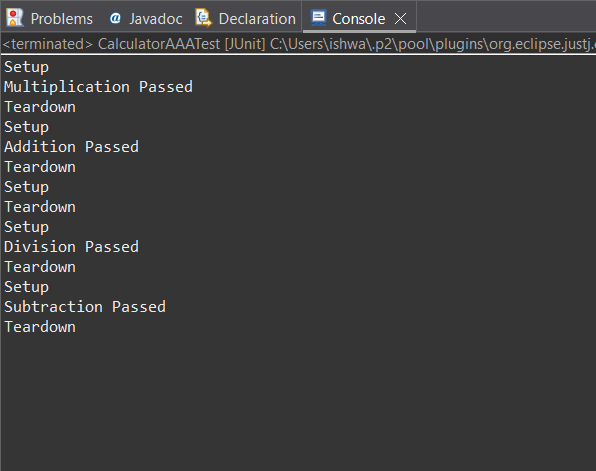
**IMPLEMENTATION IN ECLIPSE :**

****

****

**OUTPUT :**

****

****

**PROGRAM :**

package com.exam;

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 ArithmeticException("Cannot divide by zero");

return a / b;

}

}

package com.exam;

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

public class CalculatorAAATest {

private Calculator calculator;

*@Before*

public void setUp() {

calculator = new Calculator();

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

}

*@After*

public void tearDown() {

calculator = null;

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

}

*@Test*

public void testAddition() {

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

*assertEquals*(5, result);

System.***out***.println("Addition Passed");

}

*@Test*

public void testSubtraction() {

int result = calculator.subtract(5, 2);

*assertEquals*(3, result);

System.***out***.println("Subtraction Passed");

}

*@Test*

public void testMultiplication() {

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

*assertEquals*(12, result);

System.***out***.println("Multiplication Passed");

}

*@Test*

public void testDivision() {

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

*assertEquals*(5, result);

System.***out***.println("Division Passed");

}

*@Test*(expected = ArithmeticException.class)

public void testDivisionByZero() {

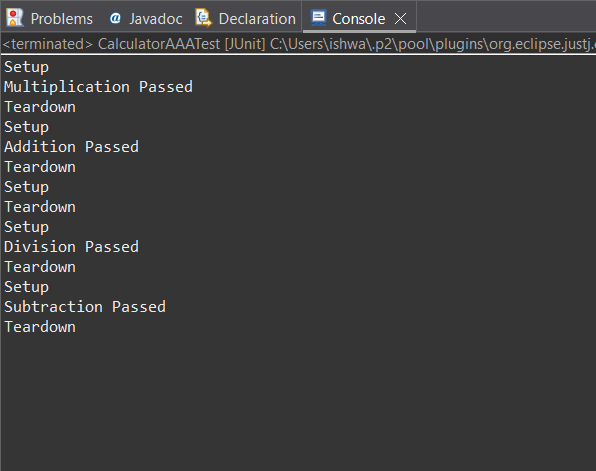
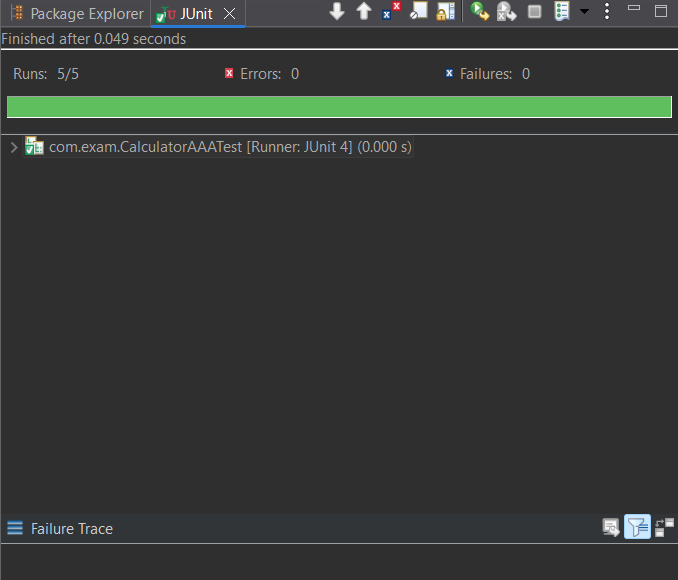
calculator.divide(5, 0);

System.***out***.println("Should not print");

}

}

**OUTPUT :**

****