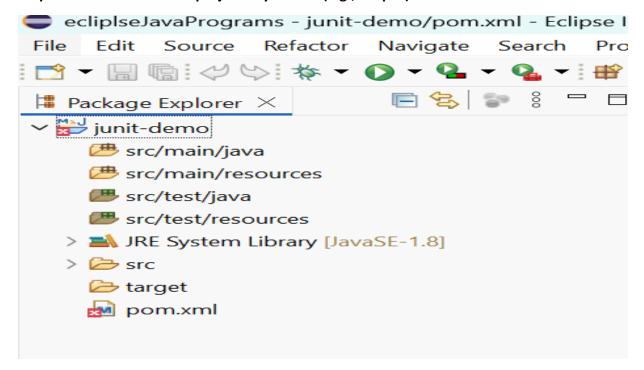
Name:- Vidya C S Track: JAVA FSE

Exercise 1: Setting Up JUnit

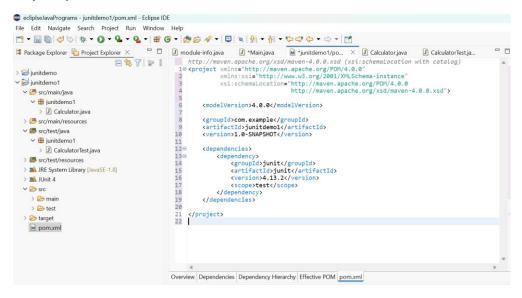
Scenario: You need to set up JUnit in your Java project to start writing unit tests.

Solution:-

Step 1:- Create a new Java project in your IDE (e.g., Eclipse).



Step 2:- Add JUnit dependency to your project. If you are using Maven, add the following to your pom.xml:



3. Create a new test class in your project.

Calculator class to test

```
ecliplseJavaPrograms - junitdemo1/src/main/java/junitdemo1/Calculator.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
| Package Explorer Project Explorer X | Project Explorer X | Package Junitdemo1;
                             ∨ ₩ junitdemo1
  ∨ # junitdemo1
    >   Calculator.iava
  > # src/main/resources
 ∨ 

src/test/java

→ 

iunitdemo1

    >  src/test/resources
 > 🛋 JRE System Library [JavaSE-1.8]
 > Numit 4
 > 🗁 main
   > 🦳 test
 > 🗁 target
   pom.xml
```

CalculatorTest class to test Calculator class

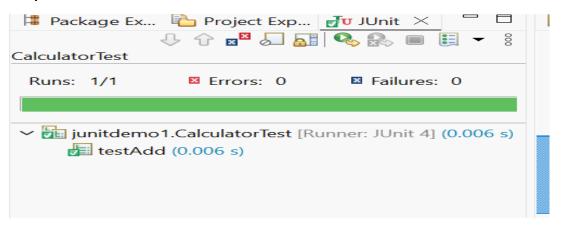
```
ecliplseJavaPrograms - junitdemo1/src/test/java/junitdemo1/CalculatorTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

The Refactor Navigate Search Project Run Window Help

The Refactor Navigate Search Project Run Window Help

The Refactor Navigate Search Project Run Window Help
                                          □ □ ☑ module-info.java ☑ *Main.java  *junitdemo1/po... ☑ *Calculator.java
package junitdemo1;
                                 □ ♣ 7 | > 8
> 📂 junitdemo
                                                       3 import static org.junit.Assert.*;
∨ 🔛 junitdemo1
   5 import org.junit.Test;
     public class CalculatorTest {
        >    Calculator.iava
   > # src/main/resources
                                                               public void testAdd() {
    Calculator calc = new Calculator();
    int result = calc.add(2, 3);
    assertEquals(5, result);
   >  src/test/resources
   > N JRE System Library [JavaSE-1.8]
   > M JUnit 4
   > 🗁 main
      > 🦳 test
   > 🗁 target
     pom.xml
```

Output of the test :-



Exercise 3: Assertions in Junit

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

```
File Edit Source Refactor Navigate Search Project Run Window Help
□ Project Exp... □ Project Exp... □ Unit × □ □ □ *Main.java ■ junitdemo1/p... □ Calculator.java □ CalculatorTe... □ AssertionsT... × **1
            1 package junitdemo1;
Finished after 0.024 seconds
                                              3⊖ import org.junit.Test;
4 import static org.junit.Assert.*;

■ Failures: 0

                                              6 public class AssertionsTest {
                                                     public void testAssertions() {
                                                         assertEquals("Sum should be 5", 5, 2 + 3);
                                                         // Assert true
assertTrue("5 should be greater than 3", 5 > 3);
                                                         // Assert false
assertFalse("5 should not be less than 3", 5 < 3);</pre>
                                                         // Assert null
Object obj1 = null;
assertNull("Object should be null", obj1);
                                                         // Assert not null
Object obj2 = new Object();
assertNotNull("Object should not be null", obj2);
                                   国译部
Failure Trace
```

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

Output:-

```
File Edit Source Refactor Navigate Search Project Run Window Help
🔋 Package Ex... 🖺 Project Exp... 🗗 Unit 🗴 📅 🗖 📓 junitdemo1/p... 📝 Calculator.java 📝 CalculatorTe... 📝 AssertionsT...
         public void tearDown() {
                                                  calculator = null;
System.out.println("Teardown: Calculator instance destroyed");
inished after 0.024 seconds
                                        24
                                        25
Runs: 2/2 

☑ Errors: 0

■ Failures: 0

                                               // ⋄ Test 1 using AAA
                                              public void testAdd() {
> iunitdemo1.CalculatorWithFixtureTest [Runner: JUnit 4
                                                  int result = calculator.add(8, 2);
                                        33
                                        34
35
                                                  assertEquals("8 + 2 should equal 10", 10, result);
                                       36
37
38
                                               // ⋄ Test 2 using AAA
                                       39 <del>-</del>
40
41
42
43
44
                                               public void testMultiply() {
                                                  int result = calculator.multiply(3, 4);
                                                  assertEquals("3 * 4 should equal 12", 12, result);
                              国产品
                                       47
```

Setup: Calculator instance created

Teardown: Calculator instance destroyed

Setup: Calculator instance created

Teardown: Calculator instance destroyed