**JUnit Testing Exercises**

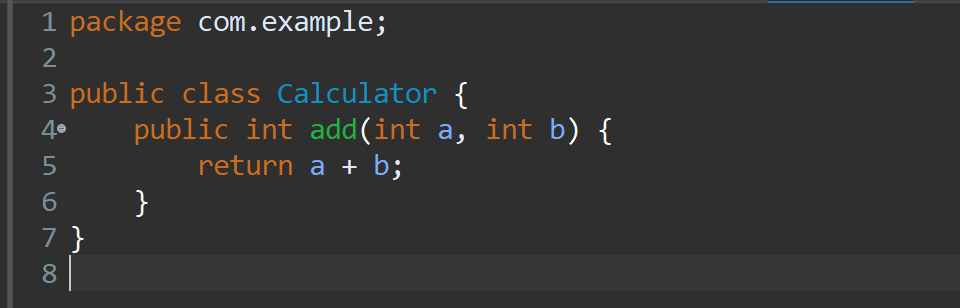
**Exercise 1: Setting Up JUnit**

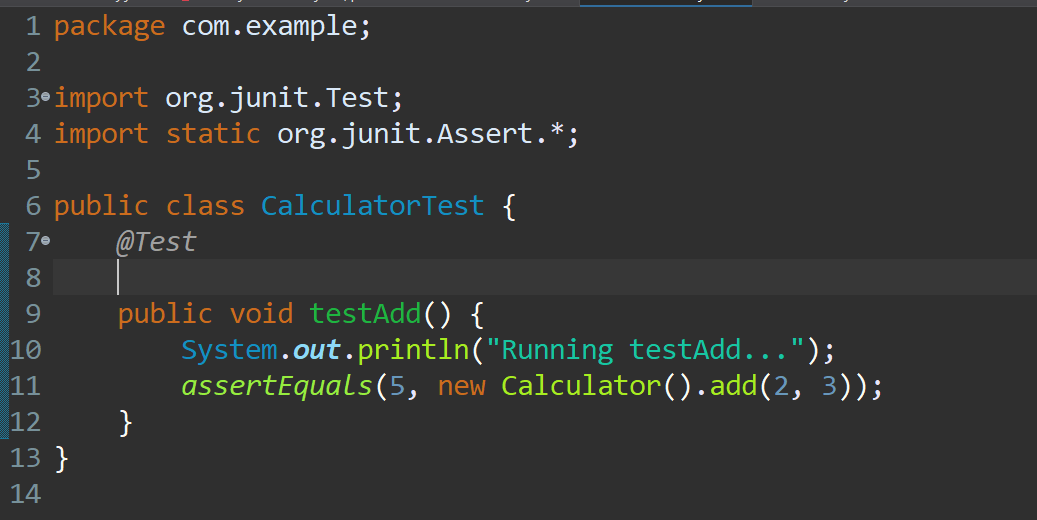
Scenario: You need to set up JUnit in your Java project to start writing unit tests.

**Step 1: Create a New Java Project**  
  
Create a Maven Project:  
Eclipse: File > New > Maven Project  
  
  
**Step 2: Add JUnit Dependency**

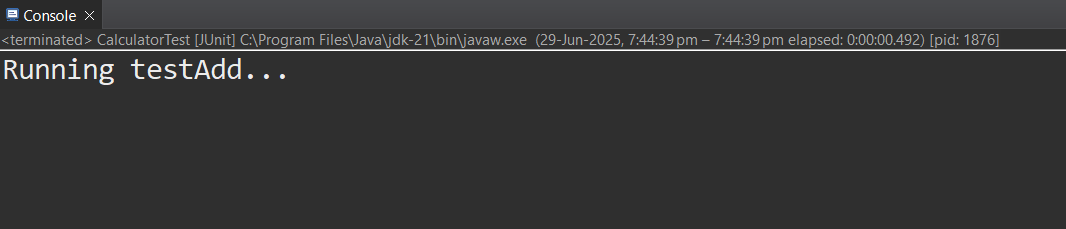
Add the following to your pom.xml inside the <dependencies> section:  
  
  
<dependencies>  
<dependency>  
<groupId>junit</groupId>  
<artifactId>junit</artifactId>  
<version>4.13.2</version>  
<scope>test</scope>  
</dependency>  
</dependencies>

**Step 3: Create a New Java Class**  
Let’s assume you're writing a simple utility class.

📁 src/main/java/com/example/Calculator.java  
  


**Step 4: Create a JUnit Test Class**  
📁 src/test/java/com/example/CalculatorTest.java  
  


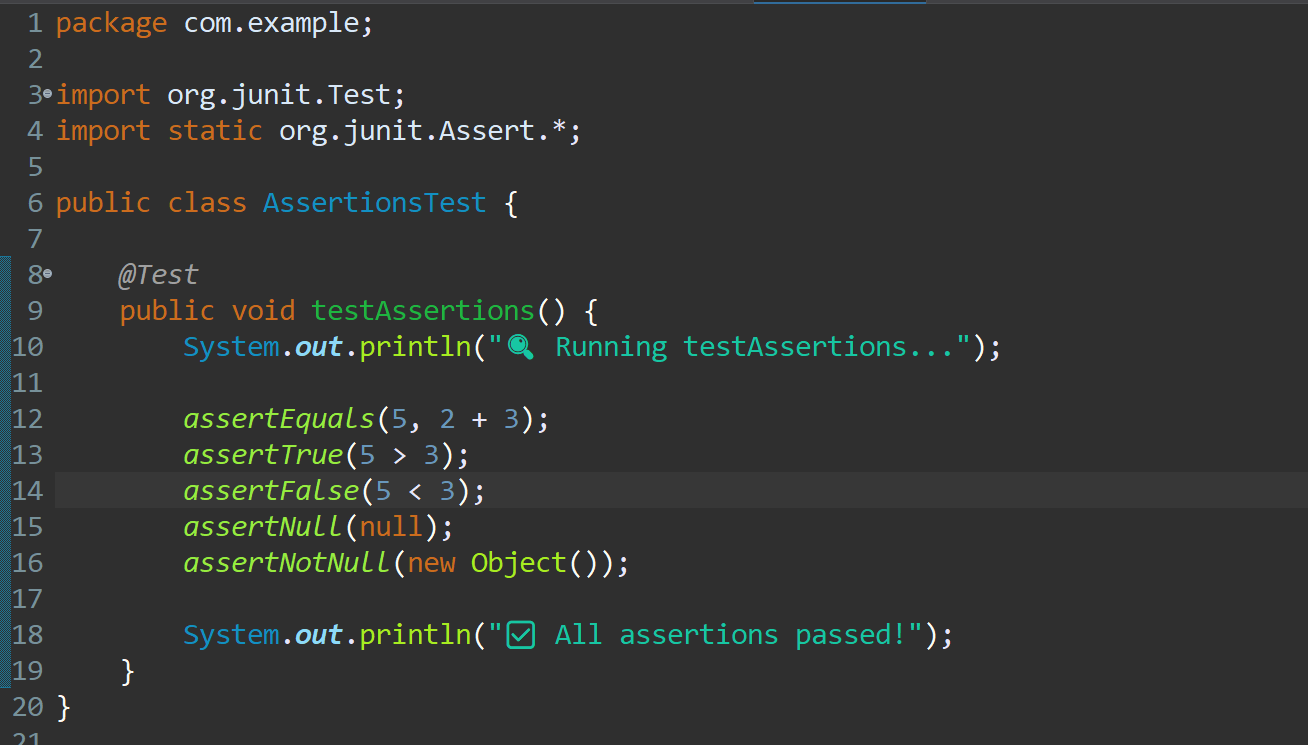
**How to Run the Test**  
  
In Eclipse: Right-click > Run As > JUnit Test  
  
 **Expected Output**



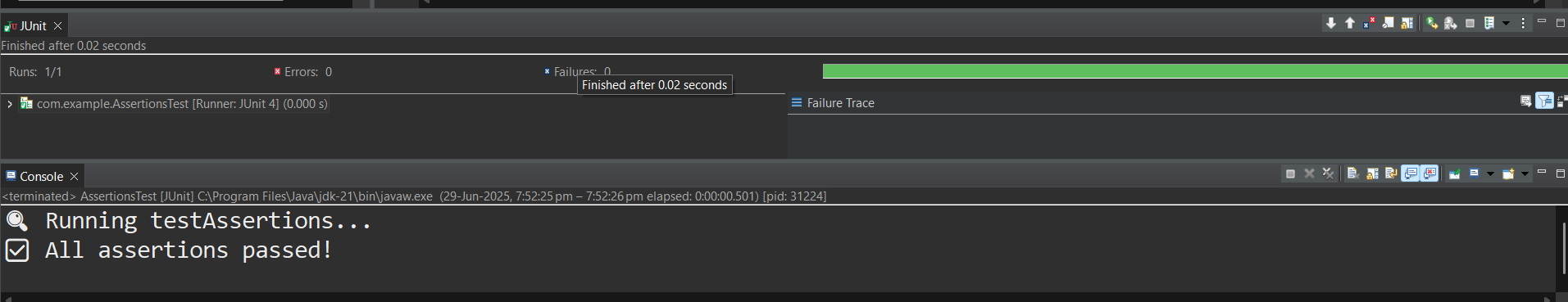
**Exercise 3: Assertions in JUnit**

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

**1. Write tests using various JUnit assertions.**



**OUTPUT:**



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

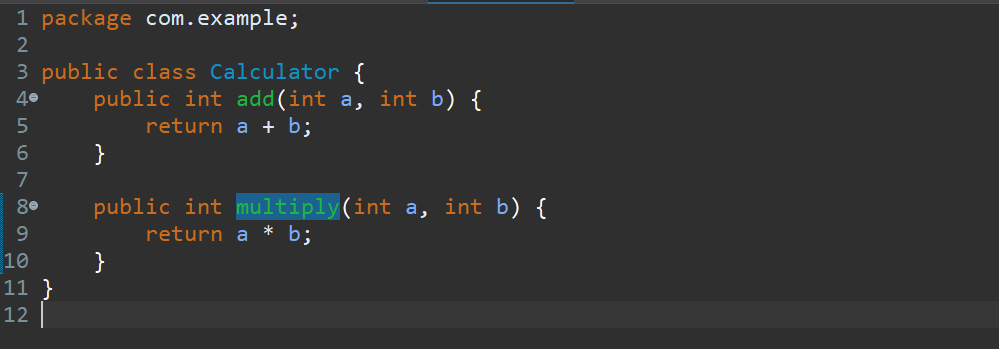
Scenario: You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

**Steps: 1. Write tests using the AAA pattern.**

### **Step 1: Create the Class Under Test**

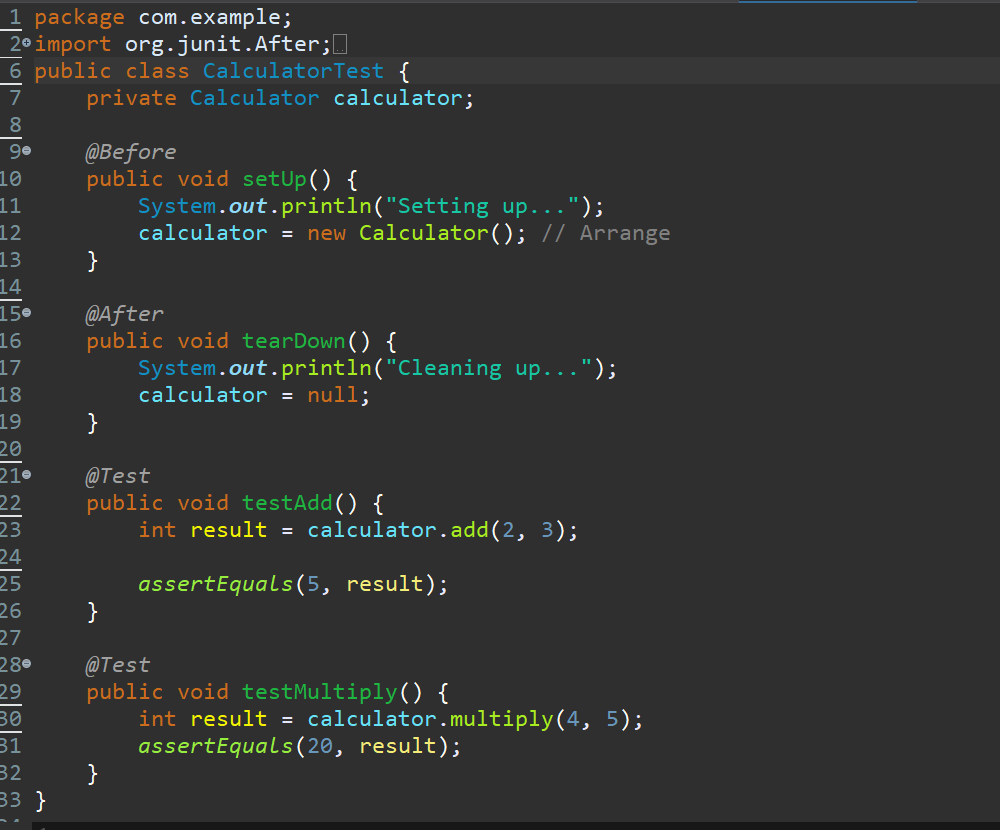
Create a simple class to test:

📁 src/main/java/com/example/Calculator.java



### **Step 2: Create the Test Class**

📁 src/test/java/com/example/CalculatorTest.java



**Run the Tests**

### **In Eclipse:**

Right-click on CalculatorTest.java → Run 'CalculatorTest'

**Expected Console Output**

