**JUNIT\_BASIC TESTING EXERCISES**

**( WEEK 2)**

**SUPER SET ID:6410152**

**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**

**In Eclipse:**

1. Open Eclipse IDE.
2. Go to File → New → Java Project.
3. Name your project (e.g., JUnitSetUpDemoNew).
4. Click Finish.

**Step 2: Manual Setup**

1. Right-click your project → Build Path → Add Libraries…

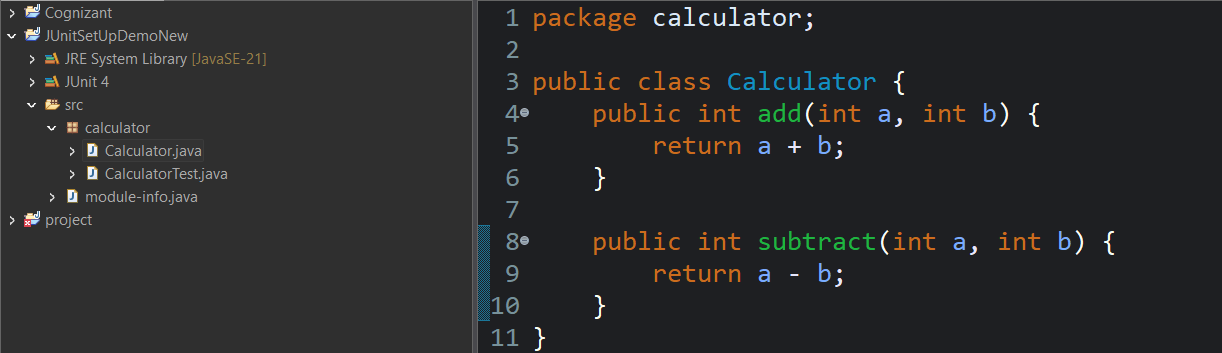
2. Choose JUnit → Next.

3. Select JUnit 4 → Click Finish.

**Step 3: Create a New Test Class**

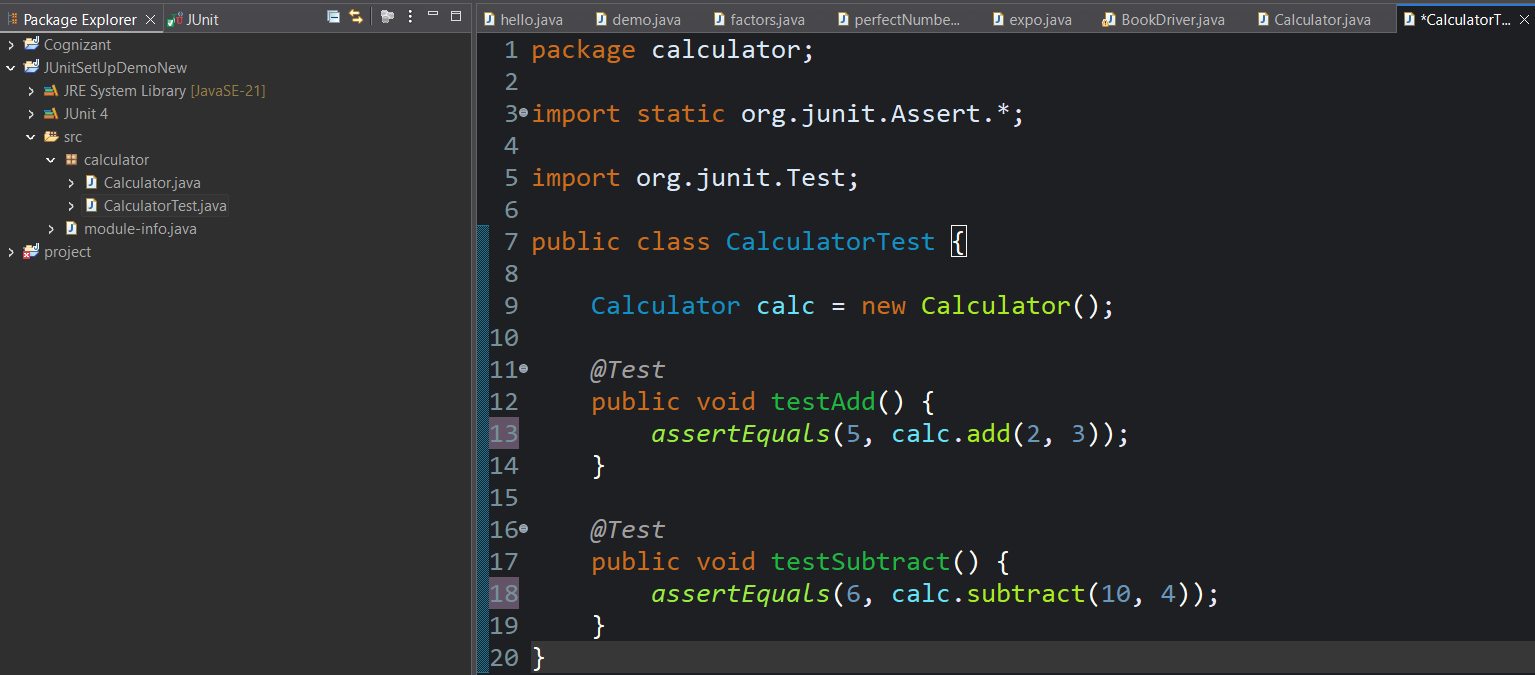
**A. Create the Main Class to Be Tested**

1. Right-click src → New → Class.
2. Name it: Calculator.
3. Add this code:



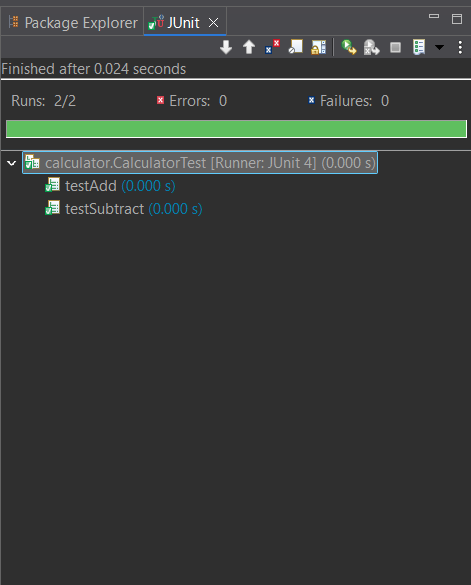
**B. Create the Test Class**

1. Right-click the src folder → New → JUnit Test Case.
2. Name it: CalculatorTest.
3. Select JUnit 4.
4. Click Finish.
5. Add this code:



**Step 4: Run the Test**

1. Right-click on CalculatorTest.java
2. Choose Run As → JUnit Test.



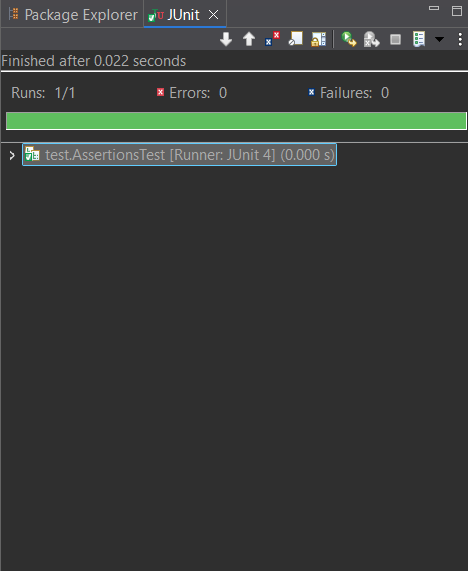
**Exercise 3: Assertions in JUnit**

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

**File Name:AssertionsTest.java**



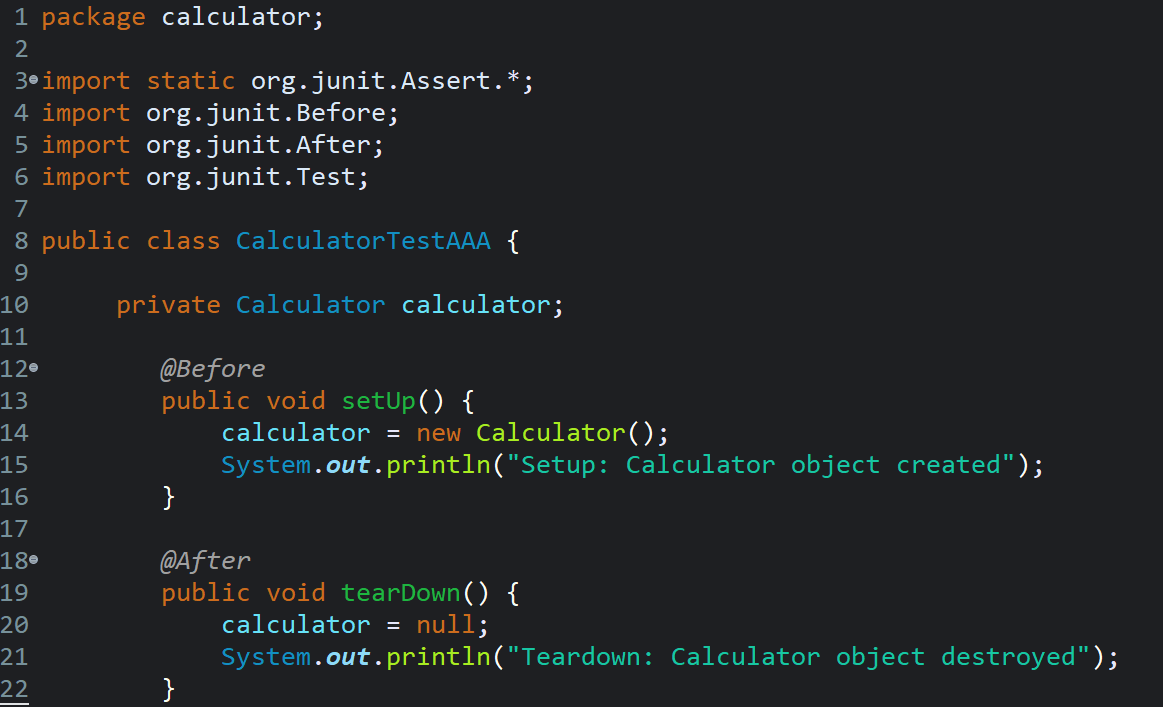
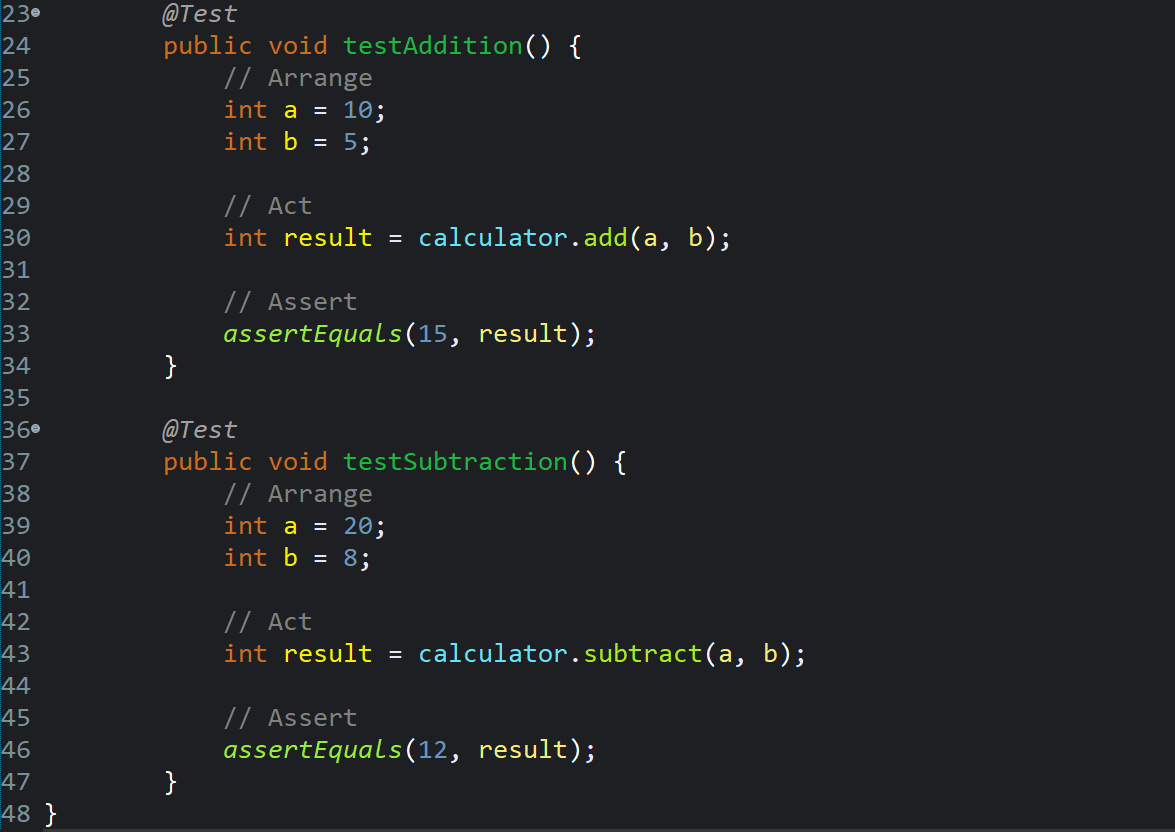
**Output:**



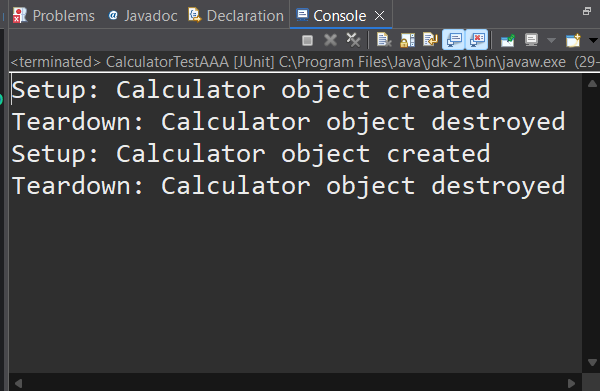
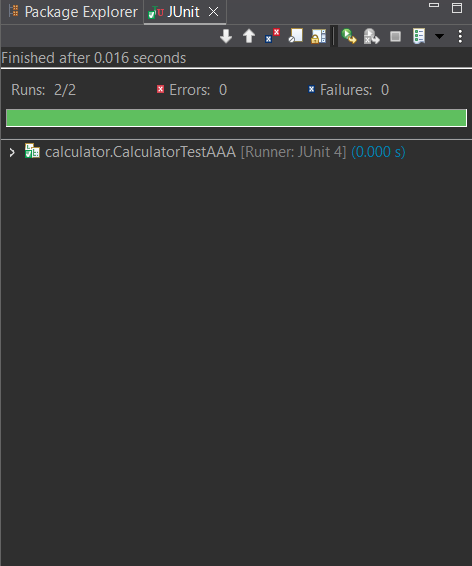
**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

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

**File Name:CalculatorTestAAA.java**

**Output:**

**SUPERSET ID:6410152**

**Exercise 1: Logging Error Messages and Warning Levels**

**Task: Write a Java application that demonstrates logging error messages and warning levels using SLF4J.**

**Step 1:Add SLF4J and Logback Dependencies**

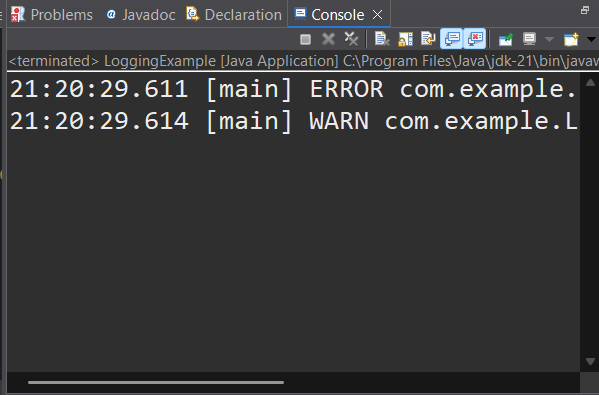
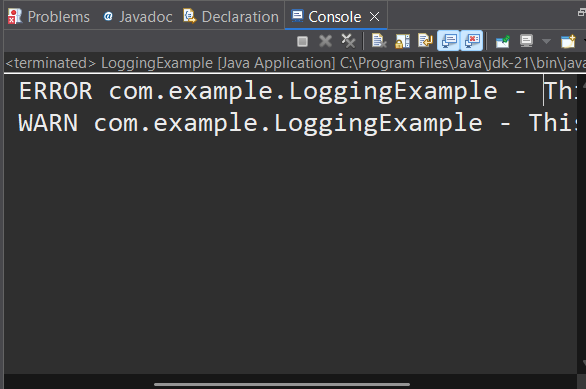
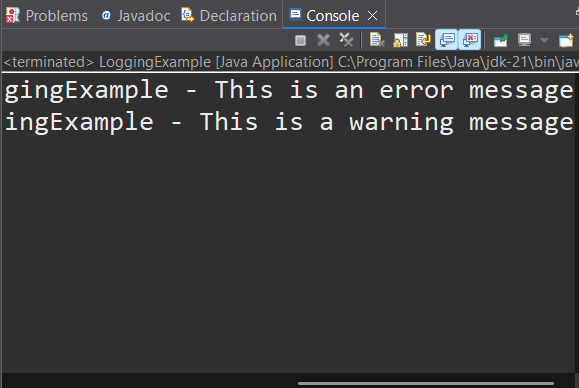
****

**Step 2:Add Logging Code**

**File Name: LoggingExample.java**

****

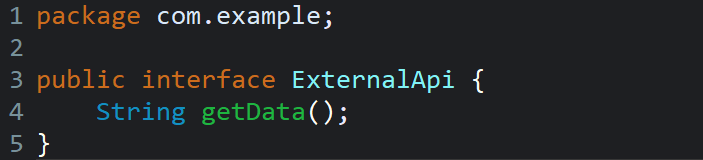
**Output:**

**  **

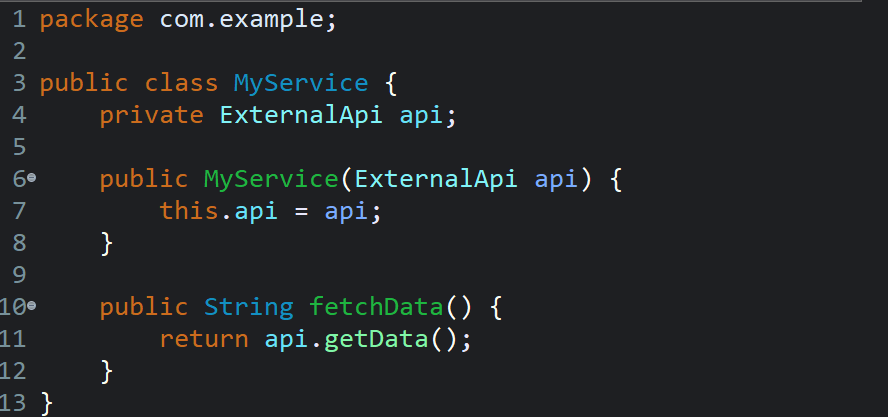
**MOCKITO HANDS-ON EXERCISES**

**Exercise 1: Mocking and Stubbing Scenario: You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.**

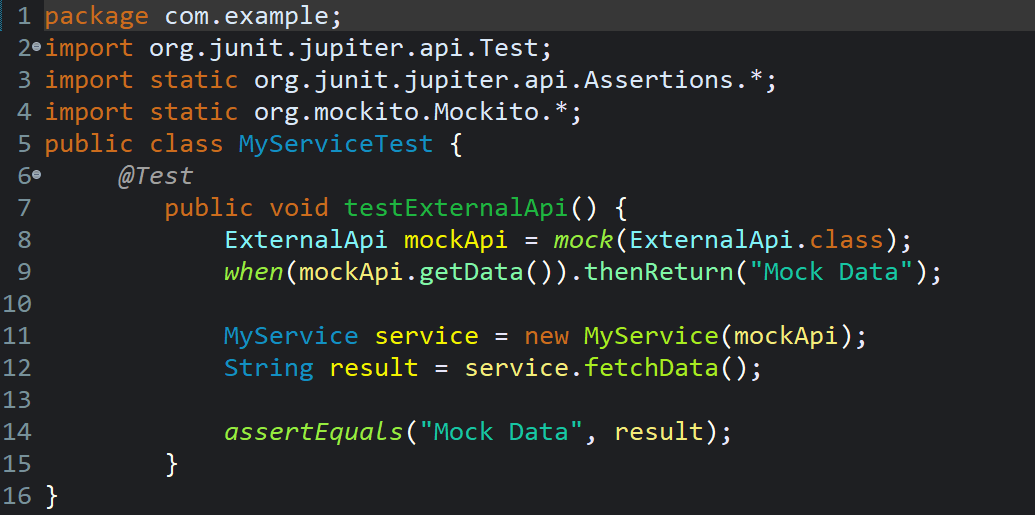
**File Name:ExternalApi.java**

****

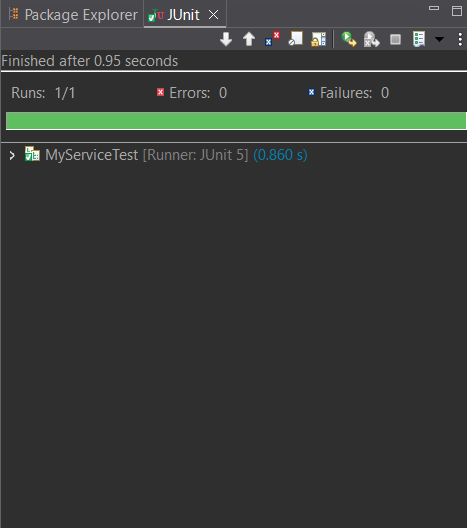
**File Name:MyService.java**

****

**File Name:MyServiceTest.java**

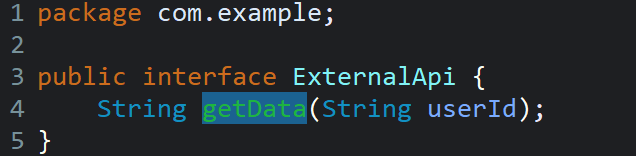
****

**Output:**

****

**Exercise 2: Verifying Interactions**

**Scenario: You need to ensure that a method is called with specific arguments.**

**File Name:ExternalApi.java**

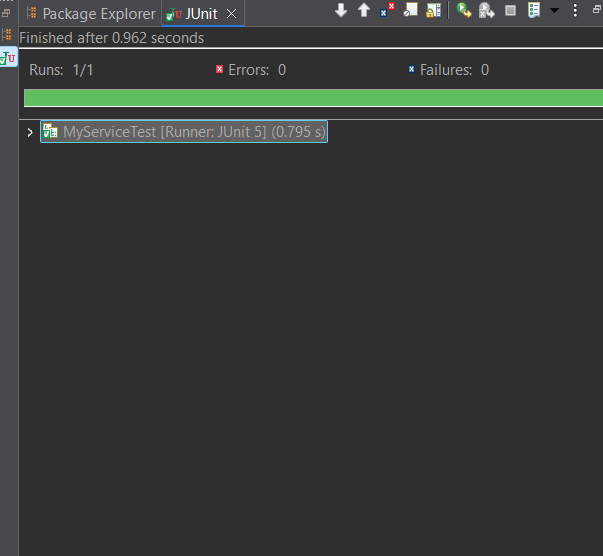
**File Name:MyService.java**

****

**File Name:MyServiceTest.java**

****

**Output:**

****