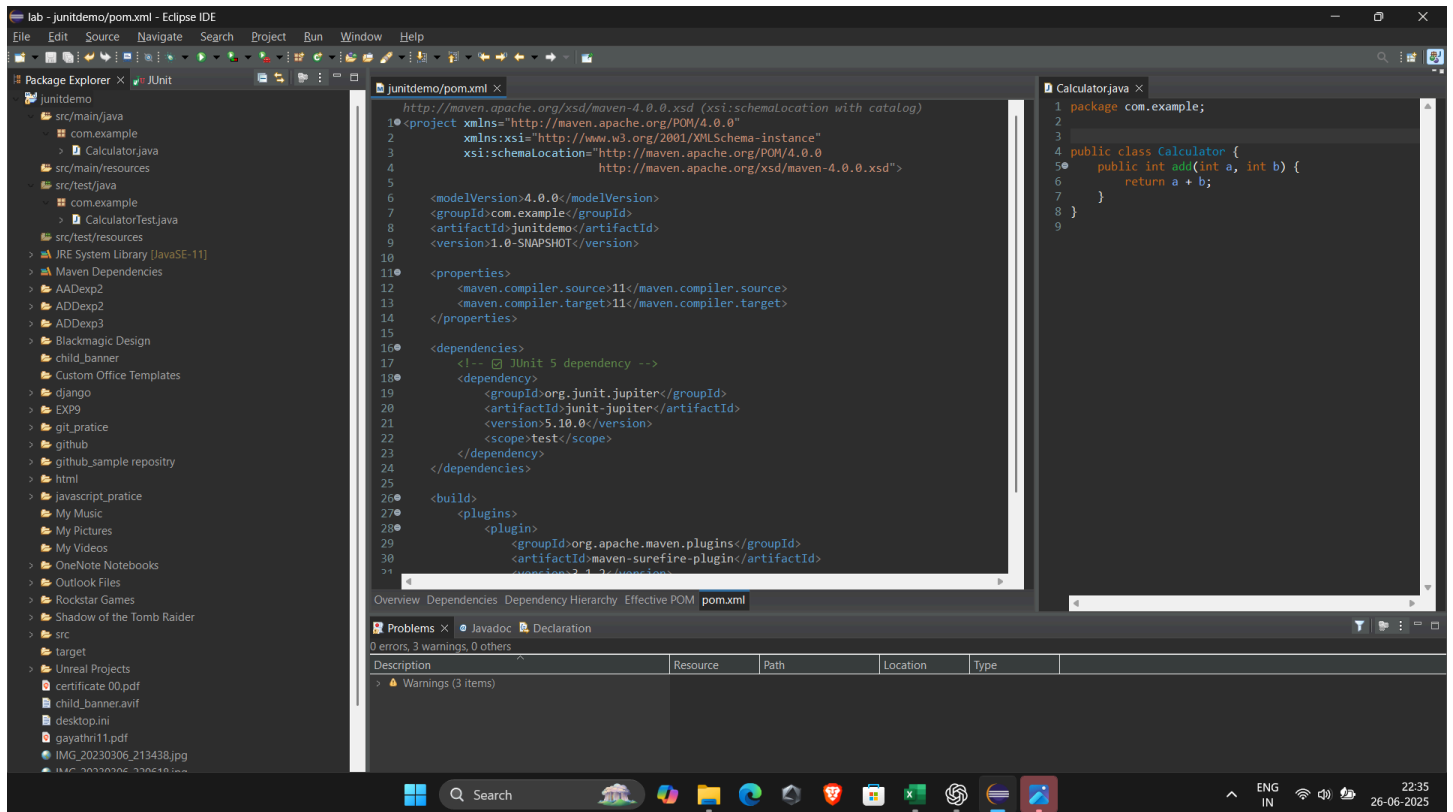
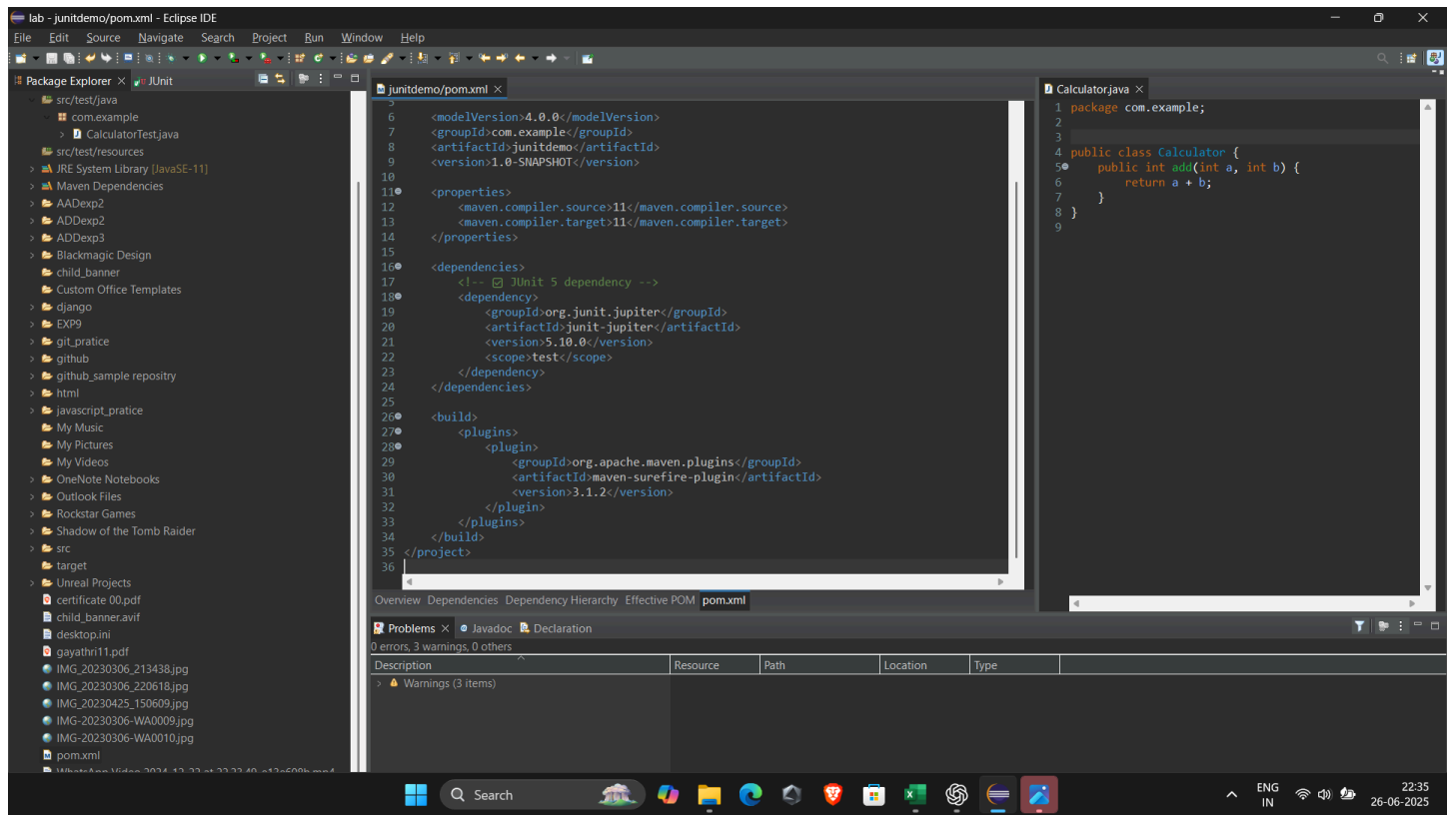


Week 2 hands on exercise solutions

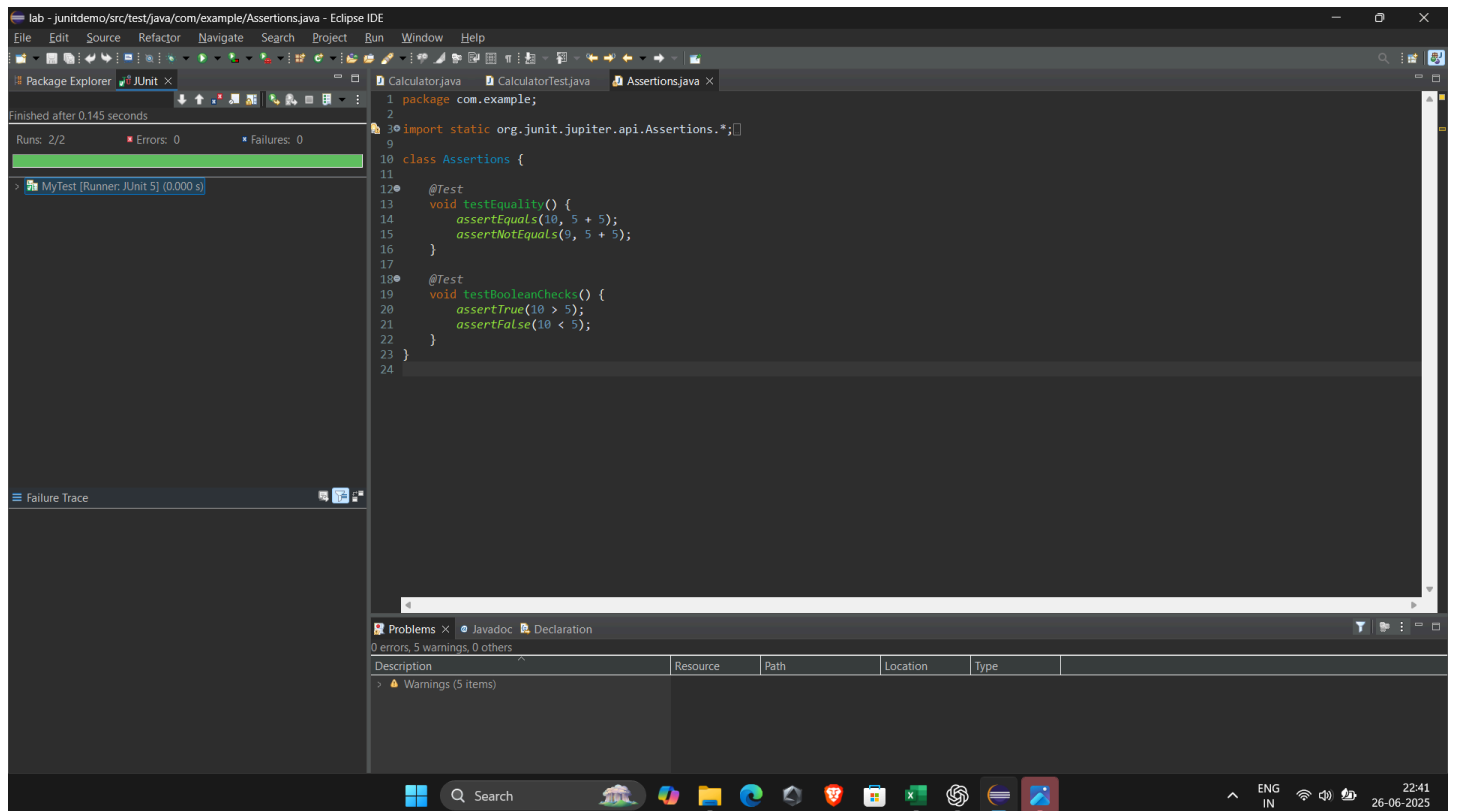
1. JUnit_Basic Testing Exercises

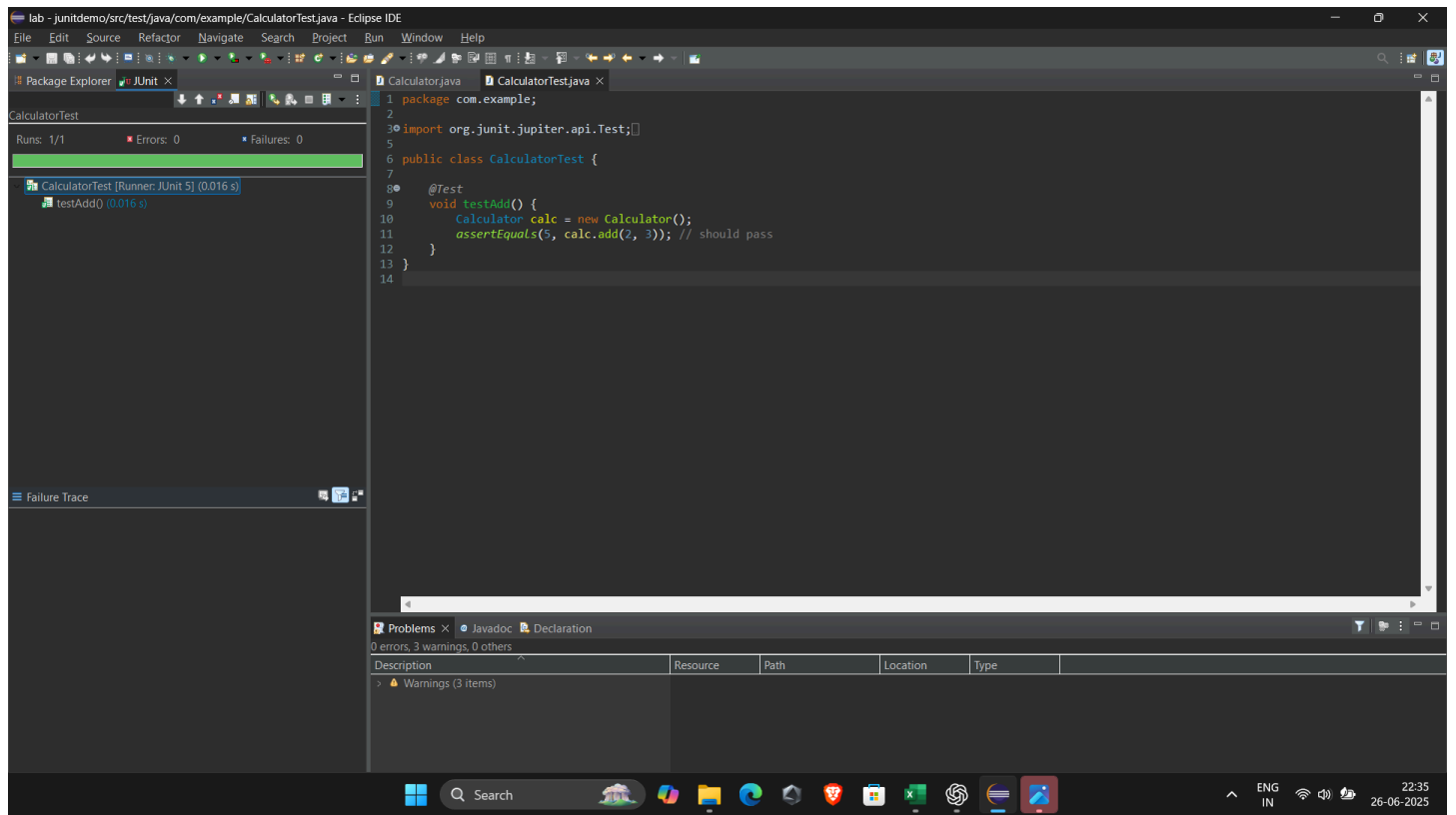
Setting Up JUnit :



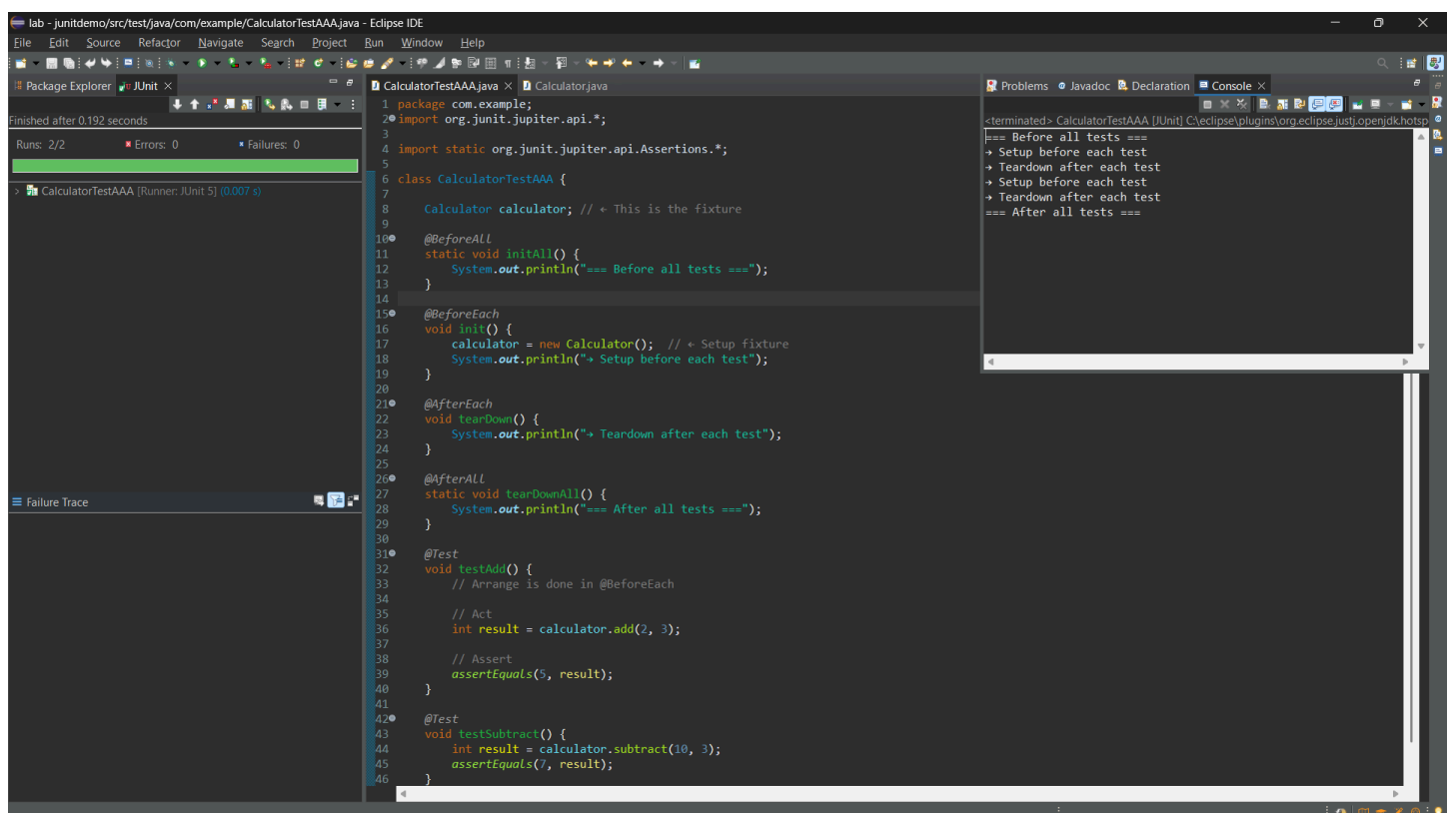


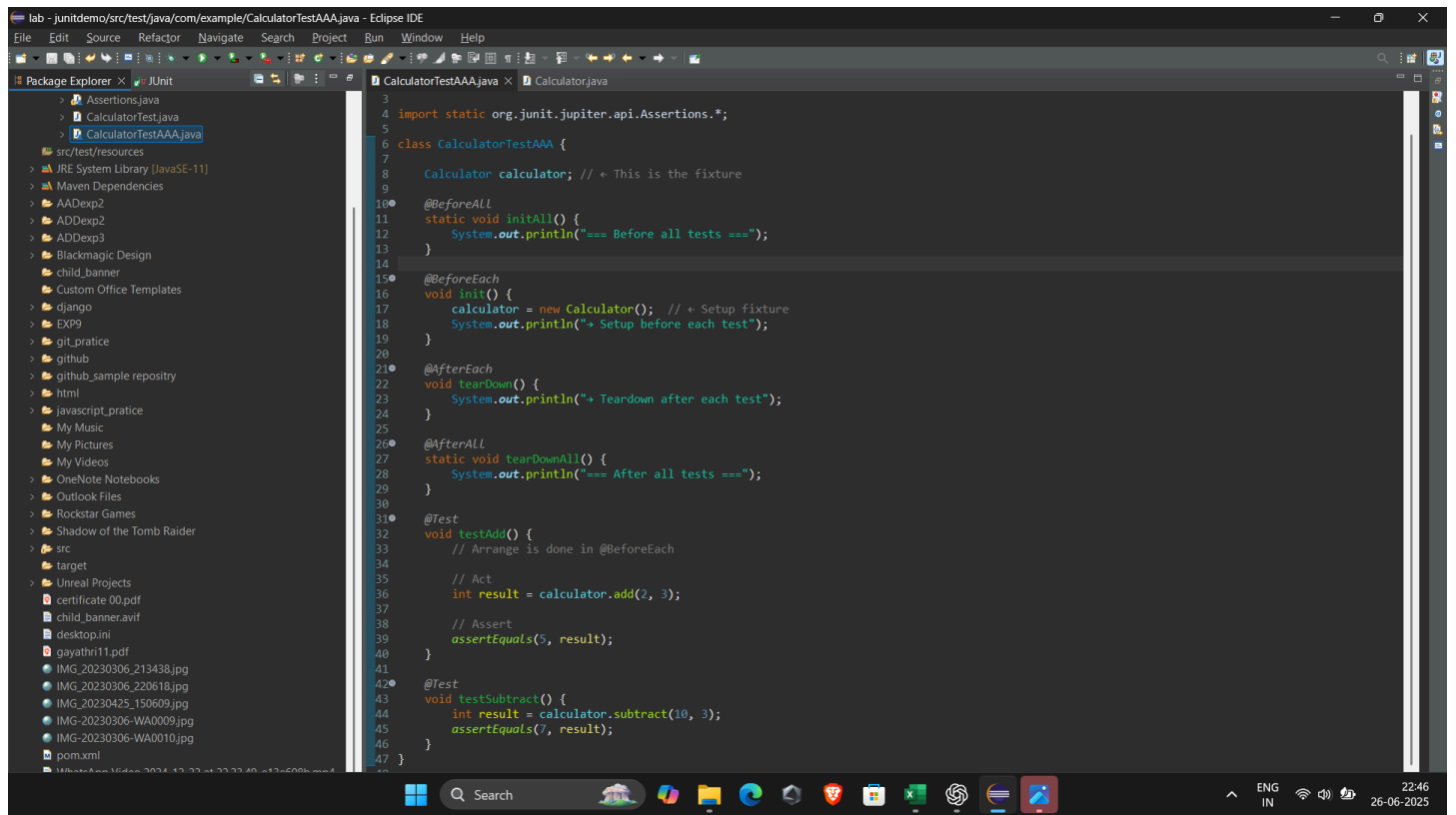
Assertions in JUnit :





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



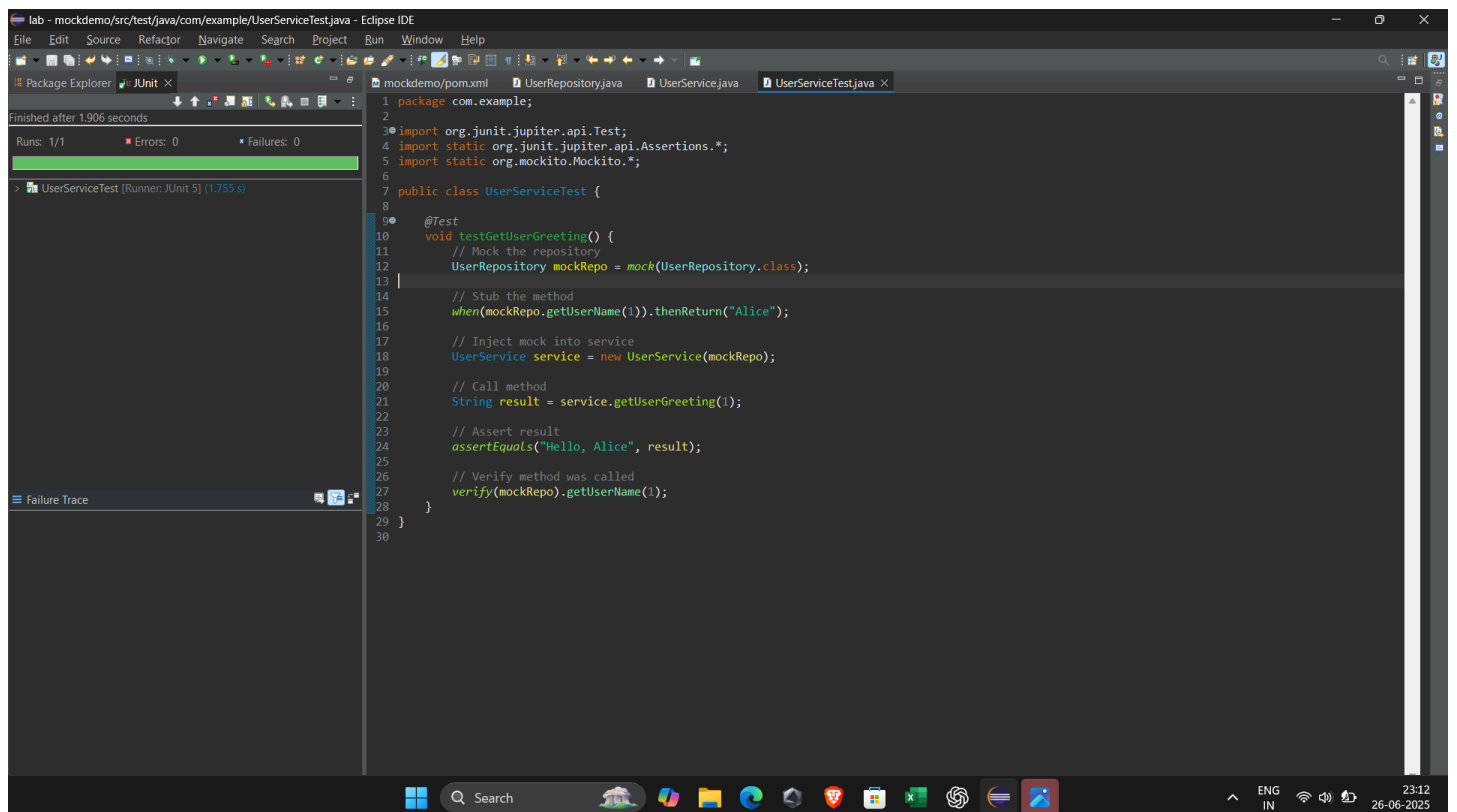


The screenshot shows the Eclipse IDE with a Java project. The Package Explorer on the left shows the project structure, including a test class `CalculatorTestAAA.java`. The main editor displays the code for `CalculatorTestAAA.java`, which uses JUnit 5 annotations for testing a `Calculator` class.

```
3
4 import static org.junit.jupiter.api.Assertions.*;
5
6 class CalculatorTestAAA {
7
8     Calculator calculator; // ← This is the fixture
9
10    @BeforeAll
11    static void initAll() {
12        System.out.println("=== Before all tests ===");
13    }
14
15    @BeforeEach
16    void init() {
17        calculator = new Calculator(); // ← Setup Fixture
18        System.out.println("→ Setup before each test");
19    }
20
21    @AfterEach
22    void tearDown() {
23        System.out.println("→ Teardown after each test");
24    }
25
26    @AfterAll
27    static void tearDownAll() {
28        System.out.println("=== After all tests ===");
29    }
30
31    @Test
32    void testAdd() {
33        // Arrange is done in @BeforeEach
34
35        // Act
36        int result = calculator.add(2, 3);
37
38        // Assert
39        assertEquals(5, result);
40    }
41
42    @Test
43    void testSubtract() {
44        int result = calculator.subtract(10, 3);
45        assertEquals(7, result);
46    }
47 }
```

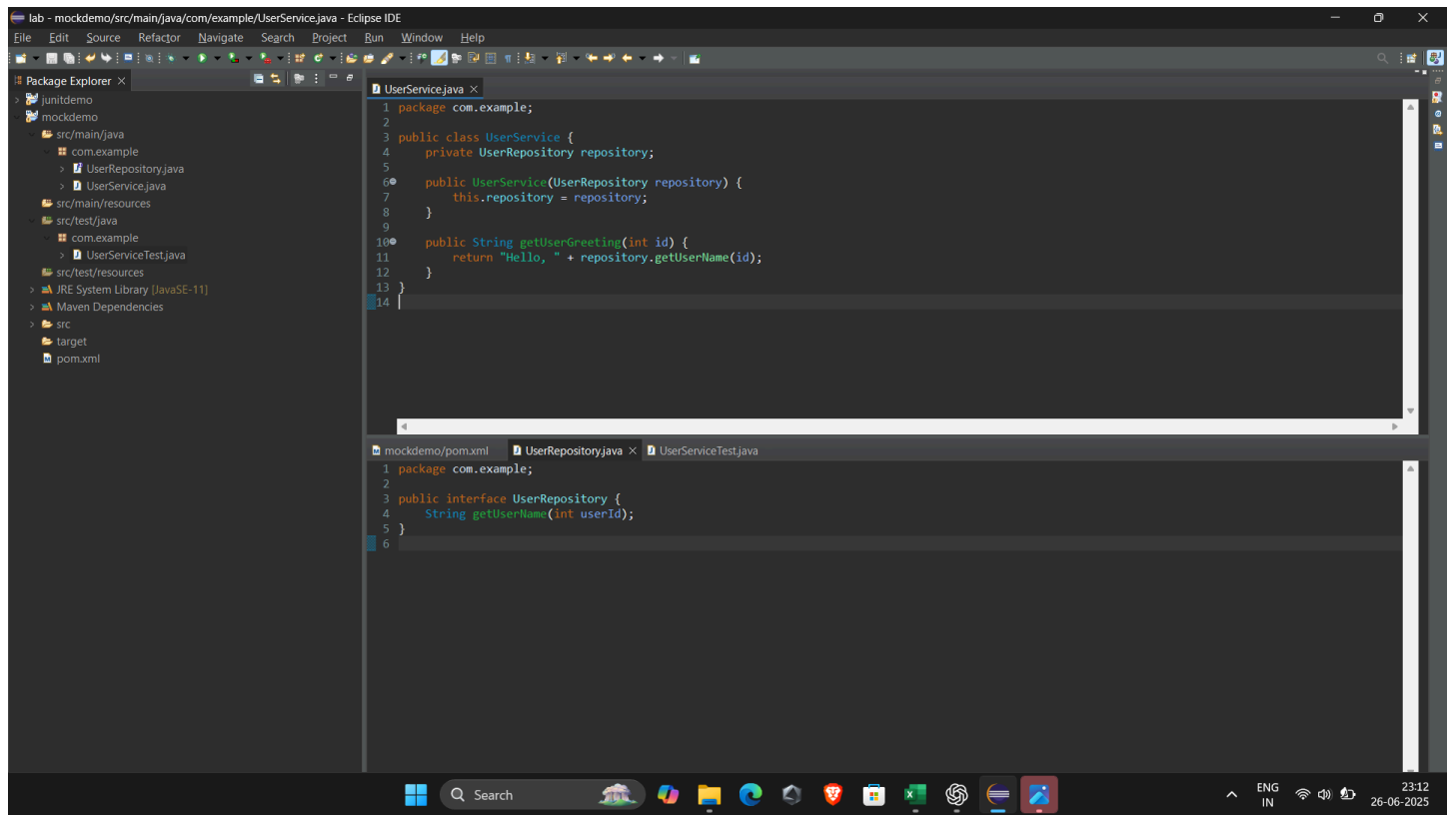
2. Mockito exercises

Mocking and Stubbing:

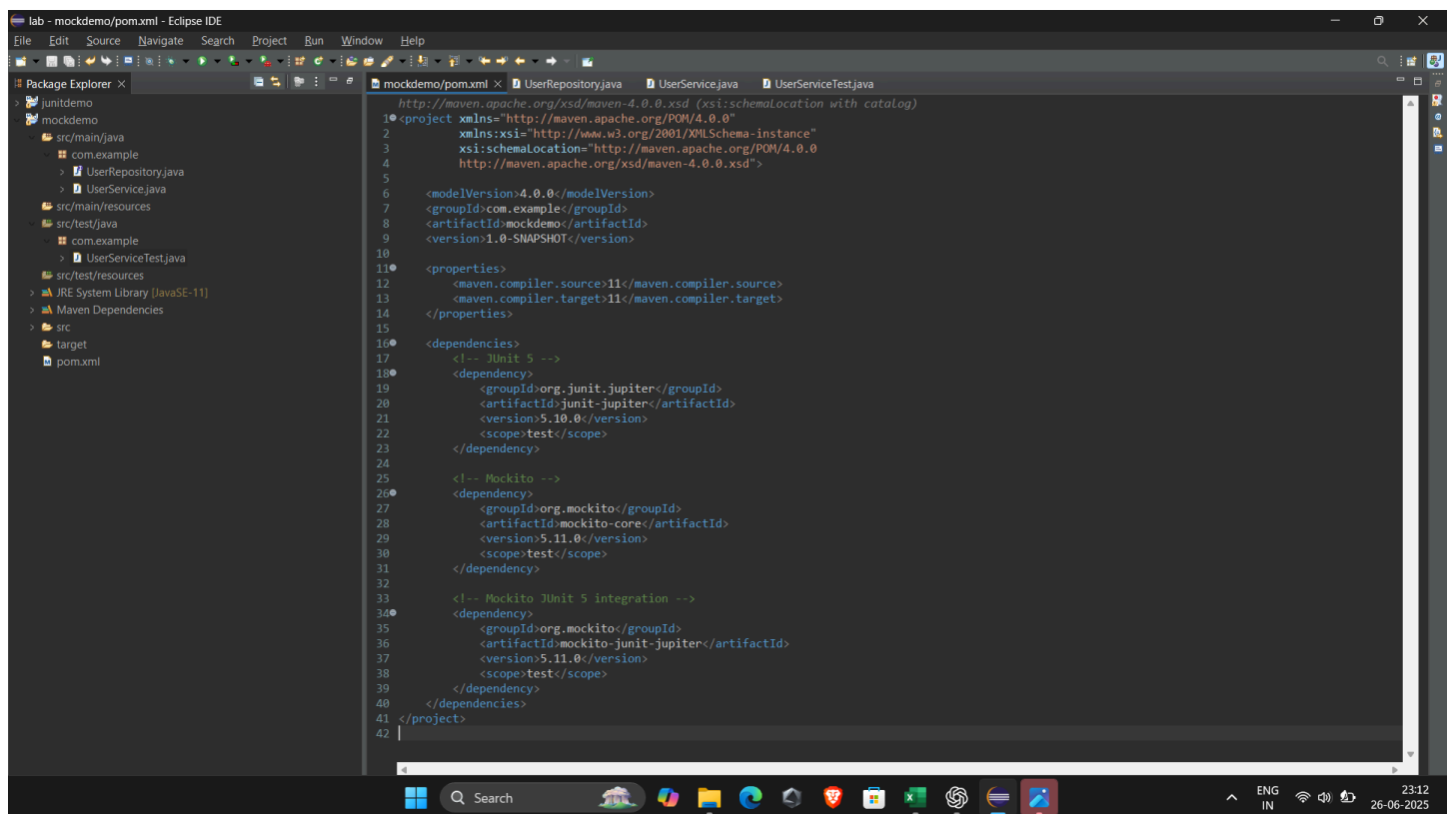


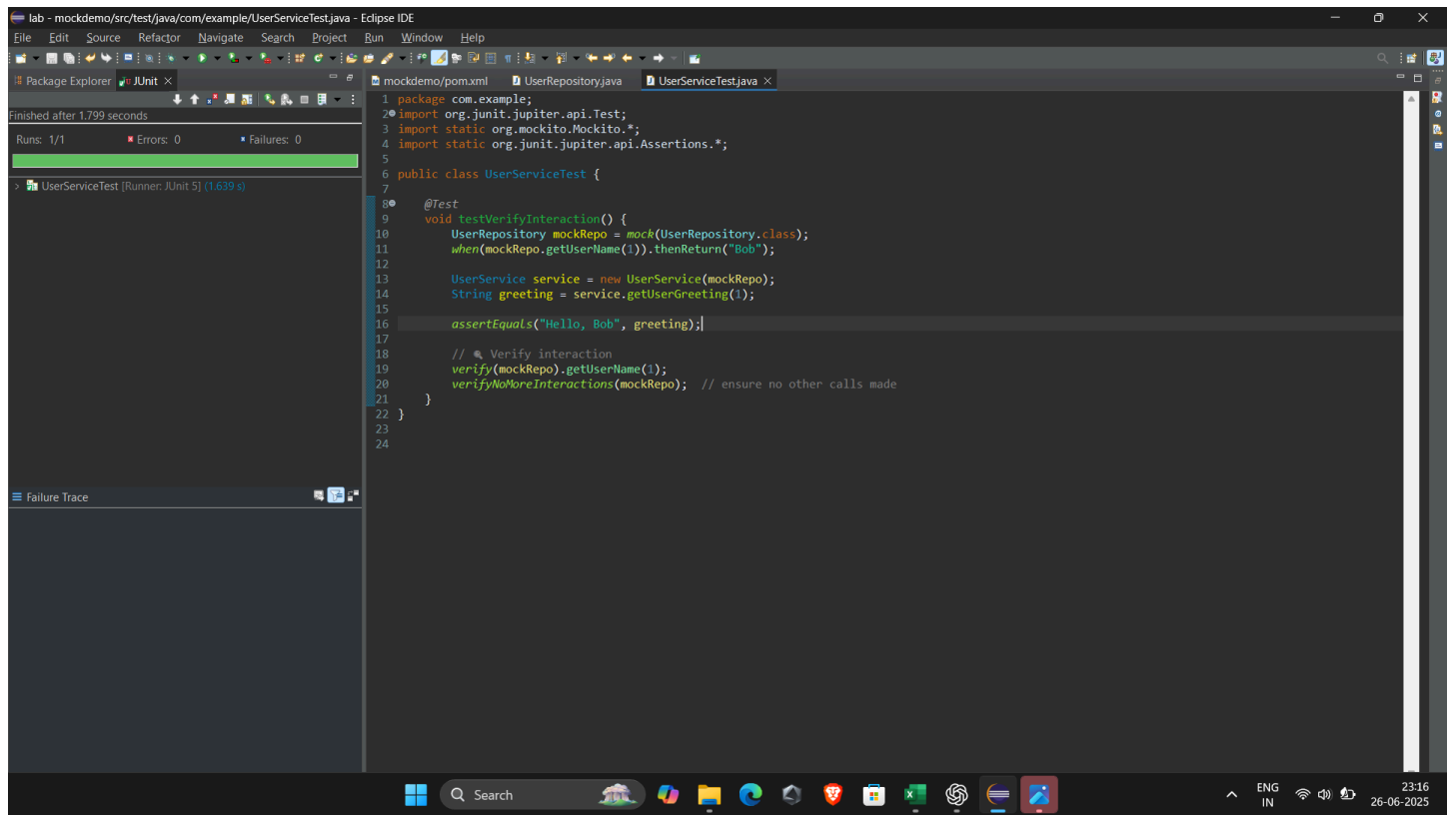
The screenshot shows the Eclipse IDE with a Java project. The Package Explorer on the left shows the project structure, including a test class `UserServiceTest.java`. The main editor displays the code for `UserServiceTest.java`, which uses Mockito for mocking and stubbing a `UserRepository` class.

```
1 package com.example;
2
3 import org.junit.jupiter.api.Test;
4 import static org.junit.jupiter.api.Assertions.*;
5 import static org.mockito.Mockito.*;
6
7 public class UserServiceTest {
8
9     @Test
10    void testGetUserGreeting() {
11        // Mock the repository
12        UserRepository mockRepo = mock(UserRepository.class);
13
14        // Stub the method
15        when(mockRepo.getUserName(1)).thenReturn("Alice");
16
17        // Inject mock into service
18        UserService service = new UserService(mockRepo);
19
20        // Call method
21        String result = service.getUserGreeting(1);
22
23        // Assert result
24        assertEquals("Hello, Alice", result);
25
26        // Verify method was called
27        verify(mockRepo).getUserName(1);
28    }
29 }
```



Verifying Interactions :





3. SL4J Logging exercises

Logging Error Messages and Warning Levels:

