

EXERCISE – 1 SETTING UP JUNIT

Add Dependency in pom.xml file

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.13.2</version>
  <scope>test</scope>
</dependency>
```

EXERCISE 3: USING ASSERTIONS IN JUNIT

```
import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

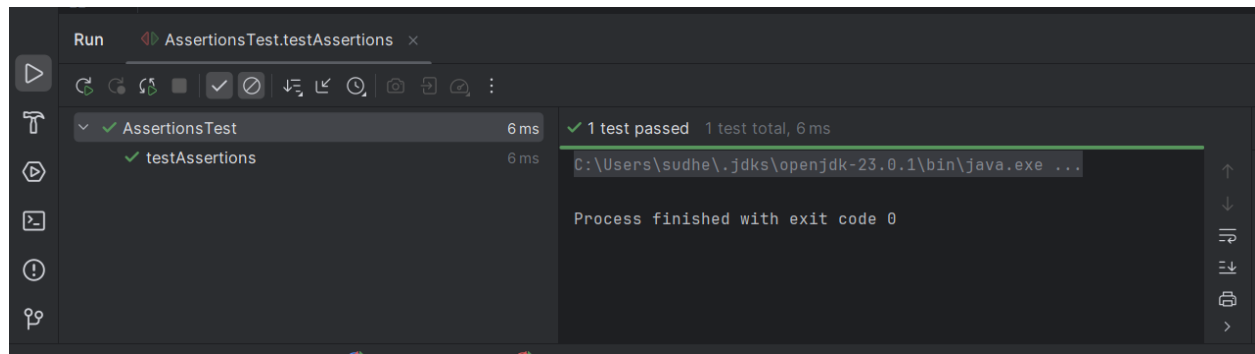
        assertFalse(5 < 3);

        assertNull(null);

        assertNotNull(new Object());
```

```
}  
}
```

Ouptut :



EXERCISE 4: AAA PATTERN + SETUP/TEARDOWN

```
import org.junit.*;
```

```
import static org.junit.Assert.*;
```

```
public class CalculatorTest {
```

```
    private Calculator calculator;
```

```
    @Before
```

```
    public void setUp(){
```

```
        calculator = new Calculator();
```

```
        System.out.println("Setup Complete");
```

```
}
```

```
@After
```

```
public void tearDown() {  
    calculator = null;  
    System.out.println("Teardown Complete");  
}
```

```
@Test
```

```
public void testAddition() {  
  
    int result = calculator.add(2, 3);  
  
    assertEquals(5, result);  
}
```

```
@Test
```

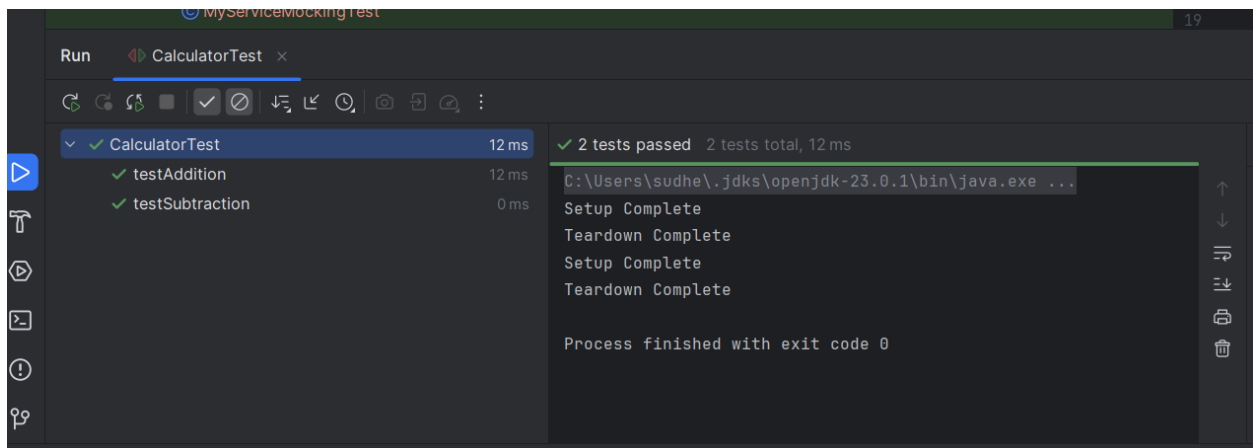
```
public void testSubtraction() {  
    int result = calculator.subtract(5, 2);  
    assertEquals(3, result);  
}
```

Calculator Class:

```
public class Calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
}
```

```
}  
  
public int subtract(int a, int b) {  
    return a - b;  
}  
}
```

Output :



EXERCISE 1: MOCKING AND STUBBING

ExternalApi.java

```
public interface ExternalApi {  
    String getData();  
}
```

MyService.java

```
public class MyService {  
    private final ExternalApi api;  
  
    public MyService(ExternalApi api) {  
        this.api = api;  
    }  
  
    public String fetchData() {  
        return api.getData();  
    }  
}
```

MyServiceMockingTest.java

```
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.assertEquals;
```

```
import static org.mockito.Mockito.*;
```

```
public class MyServiceMockingTest {
```

```
    @Test
```

```
    public void testExternalApi() {
```

```
        ExternalApi mockApi = mock(ExternalApi.class);
```

```
        when(mockApi.getData()).thenReturn("Mock Data");
```

```
        MyService service = new MyService(mockApi);
```

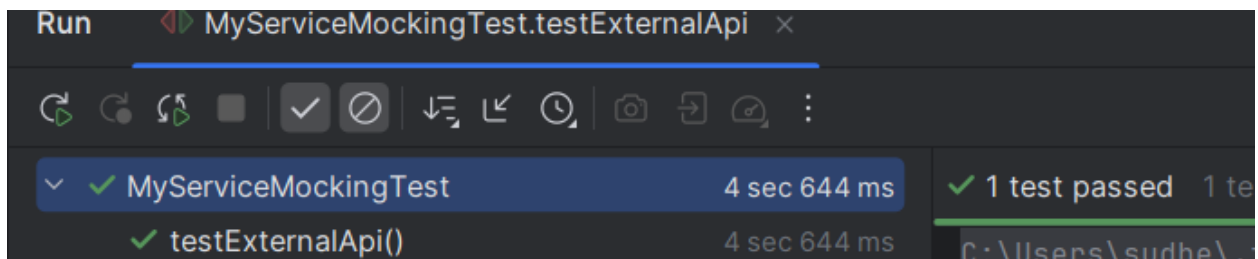
```
        String result = service.fetchData();
```

```
        assertEquals("Mock Data", result);
```

```
    }
```

```
}
```

output:



EXERCISE 2: VERIFYING INTERACTIONS

MyServiceVerifyTest.java

```
import org.junit.jupiter.api.Test;
```

```
import static org.mockito.Mockito.*;
```

```
public class MyServiceVerifyTest {
```

```
    @Test
```

```
    public void testVerifyInteraction() {
```

```
        ExternalApi mockApi = mock(ExternalApi.class);
```

```
        MyService service = new MyService(mockApi);
```

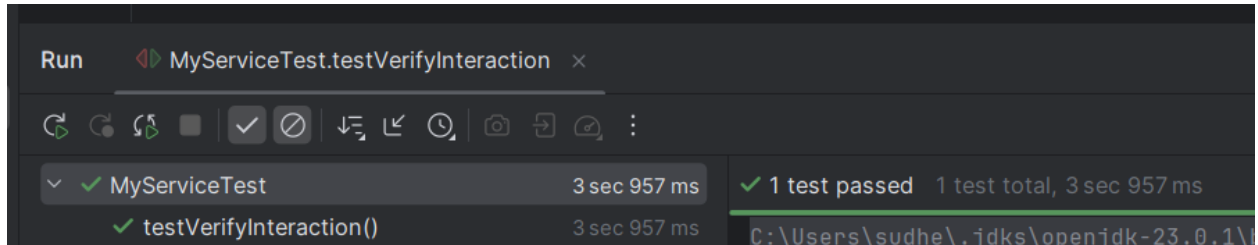
```
        service.fetchData();
```

```
        verify(mockApi).getData();
```

```
    }
```

```
}
```

Output :



EXERCISE 1: LOGGING ERROR MESSAGES AND WARNING LEVELS

Add dependency in pom.xml file

```
<dependencies>

  <dependency>

    <groupId>org.slf4j</groupId>

    <artifactId>slf4j-api</artifactId>

    <version>1.7.30</version>

  </dependency>

  <dependency>

    <groupId>ch.qos.logback</groupId>

    <artifactId>logback-classic</artifactId>

    <version>1.2.3</version>

  </dependency>

</dependencies>
```

LOGGER

```
import org.slf4j.Logger;

import org.slf4j.LoggerFactory;


public class LoggingExample {

    private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);


    public static void main(String[] args) {

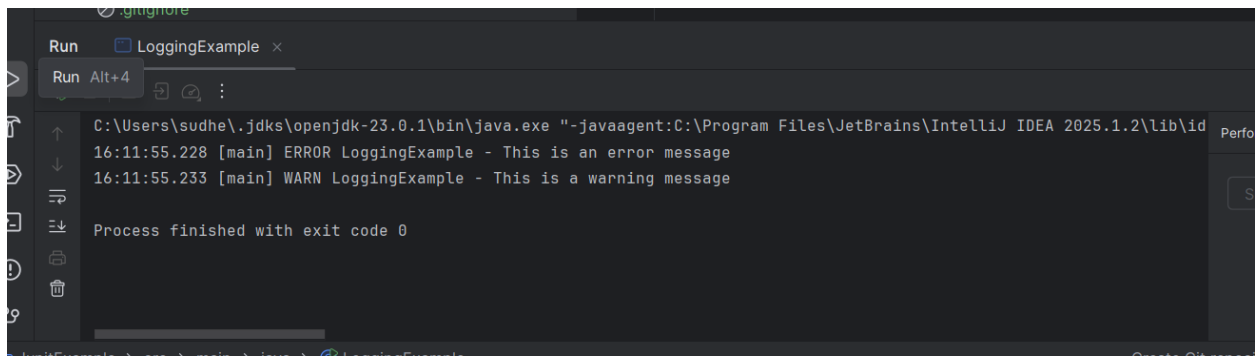
        logger.error("This is an error message");

        logger.warn("This is a warning message");

    }

}
```

Output :



The screenshot shows the Run console in IntelliJ IDEA. The title bar indicates the file is `LoggingExample`. The console output shows the execution of `java.exe` with the following messages:

```
C:\Users\sudhe\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1.2\lib\id
16:11:55.228 [main] ERROR LoggingExample - This is an error message
16:11:55.233 [main] WARN LoggingExample - This is a warning message

Process finished with exit code 0
```

The console interface includes a toolbar on the left with icons for running, stepping through code, and other debugging actions. The output is displayed in a dark-themed window.