WEEK 2

Mandatory Hands-On

**Test Driven Development**

**EX:2 Verifying Interactions**

**Dependency:**

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

**ExternalApi.java**

package org.example;

public interface ExternalApi {

String getData(String userId);

}

**MyService.java**

package org.example;

public class MyService {

private final ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchDataForUser(String userId) {

return externalApi.getData(userId);

}

}

**MyServiceTest.java**

package org.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteractionWithArgument() {

ExternalApi mockApi = mock(ExternalApi.class); when(mockApi.getData("user123")).thenReturn("Mock Data");

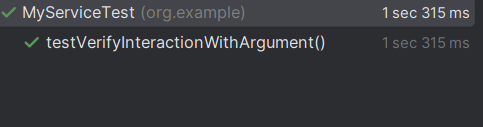
MyService service = new MyService(mockApi); service.fetchDataForUser("user123");

verify(mockApi).getData("user123");

}

}

**Output:**

****