**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.

Steps:

1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

3. Write a test case that uses the mock object.

Solution Code:

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

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

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**Source Code**

**File: ExternalApi.java**

**package** myproject.mockito;

**public** **interface** ExternalApi {

String getData();

}

**File: MyService.java**

**package** myproject.mockito;

**public** **class** MyService {

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

**File: MyServiceTest.java**

**package** myproject.mockito;

**import** **static** org.junit.jupiter.api.Assertions.assertEquals;

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

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

**public** **class** MyServiceTest {

@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);

System.***out***.println("Result from mocked API: " + result);

}

}

**Output**

Result from mocked API: Mock Data