**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);

}

}

**Answer :**

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

import org.junit.Test;

import org.mockito.Mockito;

interface ExternalApi {

String getData();

}

class MyService {

ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

public class Main {

@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);

}

public static void main(String[] args) {

Main test = new Main();

test.testExternalApi();

System.out.println("Mock test passed");

}

}

**Output :**

Mock test passed

**Exercise 2: Verifying Interactions**

Scenario:

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

Steps:

1. Create a mock object.

2. Call the method with specific arguments.

3. Verify the interaction.

Solution Code:

import static org.mockito.Mockito.\*;

import static org.junit.Assert.\*;

import org.junit.Test;

interface ExternalApi {

String getData();

}

class MyService {

ExternalApi api;

MyService(ExternalApi api) {

this.api = api;

}

String fetchData() {

return api.getData();

}

}

public class Main {

@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);

}

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

public static void main(String[] args) {

Main test = new Main();

test.testExternalApi();

test.testVerifyInteraction();

System.out.println(“Mockito tests passed");

}

}

**Output :**

Mockito tests passed