**Mockito**

**Exercise – 1: Mocking and Stubbing**

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

// External API interface to mock

interface ExternalApi {

String getData();

}

// Service that uses the External API

class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

// Test class

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);

}

}

**Output**:

MyServiceTest

✓ testExternalApi

BUILD SUCCESSFUL in 1s

1 test completed, 0 failed

**Exercise – 2: Verifying Interactions**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

interface ExternalApi {

String getData();

}

class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

// Test class

public class MyServiceTest

@Test

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

}

//act

service.fetchData();

//verify

verify(mockApi).getData();

**Ouput:**

MyServiceTest

✓ testVerifyInteraction

BUILD SUCCESSFUL in 1s

1 test completed, 0 failed