**Exercise 1: Mocking and Stubbing**

1. Project Dependencies (pom.xml)

<dependencies>

    <!-- JUnit 5 -->

    <dependency>

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

        <artifactId>junit-jupiter</artifactId>

        <version>5.9.3</version>

        <scope>test</scope>

    </dependency>

    <!-- Mockito -->

    <dependency>

        <groupId>org.mockito</groupId>

        <artifactId>mockito-core</artifactId>

        <version>4.11.0</version>

        <scope>test</scope>

    </dependency>

</dependencies>

</dependencies>

**External API Interface – ApiClient.java**

public interface ApiClient {

String fetchInfo();

}

**3. Service Layer – DataService.java**

public class DataService {

private ApiClient client;

public DataService(ApiClient client) {

this.client = client;

}

public String retrieveData() {

return client.fetchInfo();

}

}

**4. Unit Test with Stubbing – DataServiceTest.java**

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class DataServiceTest {

@Test

public void testStubbingWithMock() {

ApiClient mockClient = mock(ApiClient.class);

when(mockClient.fetchInfo()).thenReturn("Stubbed Response");

DataService service = new DataService(mockClient);

String output = service.retrieveData();

assertEquals("Stubbed Response", output);

}

}

**Exercise 2: Verifying Interactions**

**1. Interface – ApiClient.java**

public interface ApiClient {

String fetchInfo();

}

**2. Service Class – DataService.java**

public class DataService {

private ApiClient client;

public DataService(ApiClient client) {

this.client = client;

}

public String retrieveData() {

return client.fetchInfo();

}

}

**3. Unit Test for Interaction – DataServiceTest.java**

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class DataServiceTest {

@Test

public void testApiCallVerification() {

ApiClient mockClient = mock(ApiClient.class);

DataService service = new DataService(mockClient);

service.retrieveData();

verify(mockClient).fetchInfo();

}

}