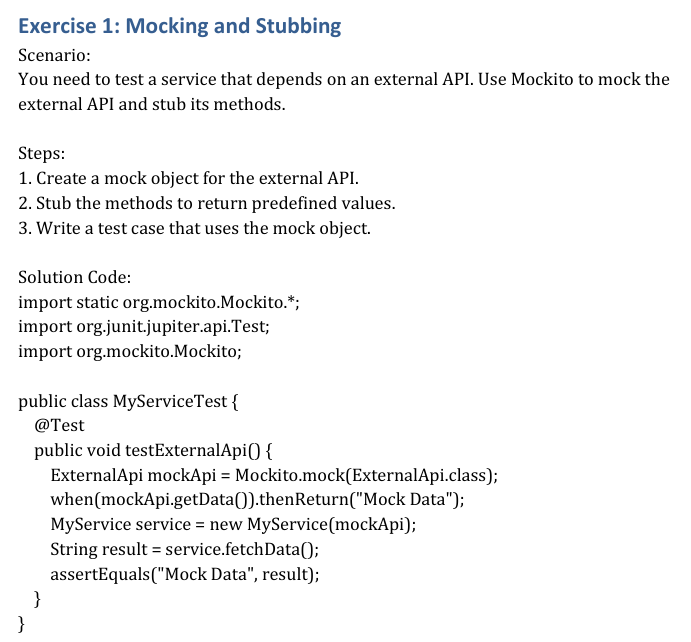
MOCKITO EXCERSICES



SOLUTION :

**Exercise 1: Mocking and Stubbing**

**Scenario: Test Service Using Mockito Mock and Stub**

**Solution Code:**

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Arrange: Create mock and stub

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

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

// Act: Use mock in service

MyService service = new MyService(mockApi);

String result = service.fetchData();

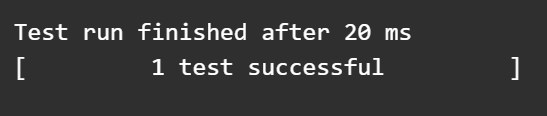
// Assert: Verify result

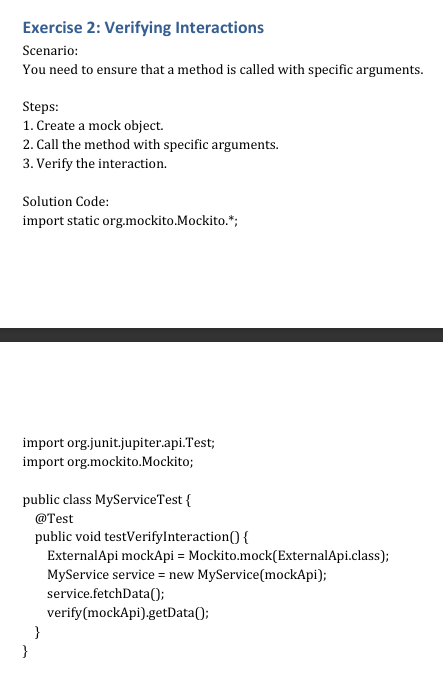
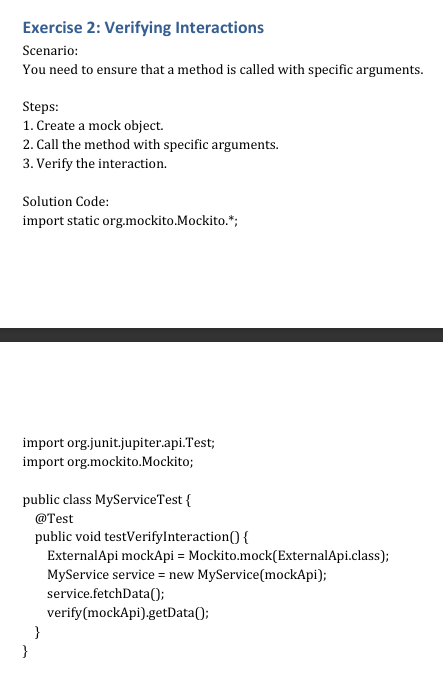
assertEquals("Mock Data", result);

}

}

**Sample Output:**





**Exercise 2: Verifying Interactions**

**Scenario: Ensure Method is Called with Specific Arguments**

**Solution Code:**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Arrange: Create mock

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

// Act: Call method

MyService service = new MyService(mockApi);

service.fetchData();

// Assert: Verify interaction

verify(mockApi).getData();

}

}

**Sample Output:**

