**WEEK – 2**

**Mockito exercises**

**Exercise 1: Mocking and Stubbing**

**CODE:**

**Pom.xml:**

<dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.9.3</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.9.3</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Mockito -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>5.6.0</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>

**ExternalApi.java:**

package com.junitdemo;  
  
public interface ExternalApi {  
 String getData();  
}

**MyService.java:**

package com.junitdemo;  
  
public class MyService {  
 private ExternalApi externalApi;  
  
 public MyService(ExternalApi externalApi) {  
 this.externalApi = externalApi;  
 }  
  
 public String fetchData() {  
 return externalApi.getData();  
 }  
}

**MyServiceTest.java:**

package com.junitdemo;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
import static org.mockito.Mockito.\*;  
  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 // Step 1: Mock ExternalApi  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
  
 // Step 2: Stub the getData method to return "Mock Data"  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 // Step 3: Inject mock into MyService and test  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 // Step 4: Assert the result  
 *assertEquals*("Mock Data", result);  
 }  
}

**OUTPUT:**



