WEEK-2

(Implemented in Eclipse)

1)Mockito

Created a Maven project named `MockitoDemo` in Eclipse and added JUnit 5 and Mockito dependencies in the `pom.xml` to enable unit testing and mocking.

pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>JUnitDemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.2</version>
        <scope>test</scope>
    </dependency>
</dependencies>
</project>
```

Exercise 1:Mocking and Stubbing:

1. Create a mock object for the external API.

   ### ExternalApi.java (Interface)

   ```java
   public interface ExternalApi {
   String getData();
   }
   ```

2. Stub the methods to return predefined values.

   ### MyService.java

   ```java
   public class MyService {
   private ExternalApi api;

   public MyService(ExternalApi api) {
     this.api = api;
   }

   public String fetchData() {
     return api.getData();
   }

   }
   ```

3. Write a test case that uses the mock object.

MyServiceTest.java

```java
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() {
                // Step 1: Mock the ExternalApi
                ExternalApi mockApi = Mockito.mock(ExternalApi.class);

                // Step 2: Stub its method
                when(mockApi.getData()).thenReturn("Mock Data");

                // Step 3: Use the mock in the service
                MyService service = new MyService(mockApi);
                String result = service.fetchData();

                // Step 4: Assert result
                assertEquals("Mock Data", result);
        }

}
```
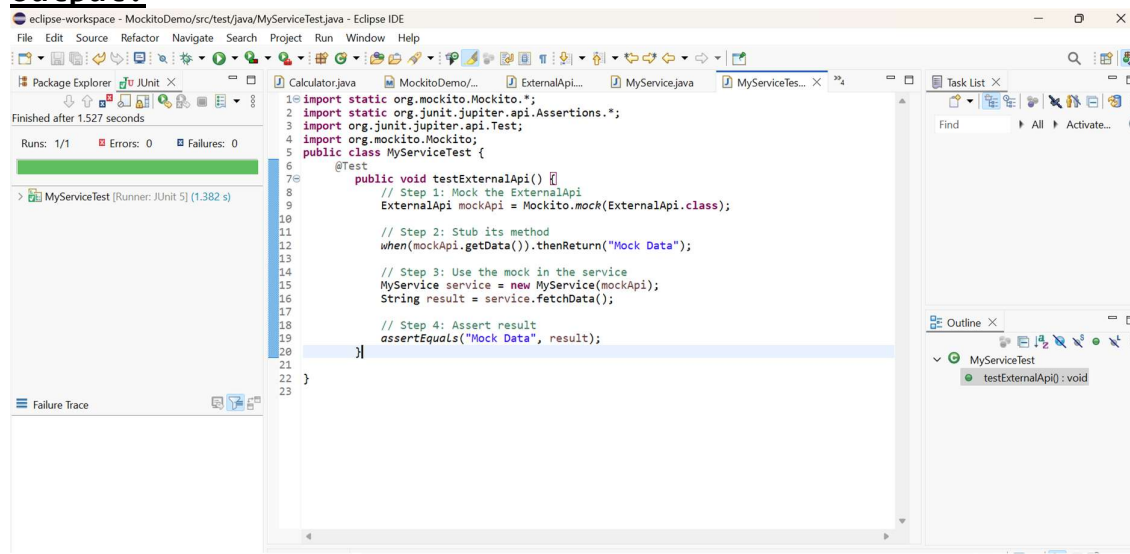
Output:

SHRAVYA VODNALA

Exercise 2: Verifying Interactions:

MyServiceTest2.java

```java
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
import org.mockito.Mockito;


public class MyServiceTest2 {
        @Test
            public void testVerifyInteraction() {
                ExternalApi mockApi = Mockito.mock(ExternalApi.class);
                MyService service = new MyService(mockApi);

                // Call the method
                service.fetchData();

                // Verify that getData() was called
                verify(mockApi).getData();
            }

}
```

Output:

SHRAVYA VODNALA