```
----------pom.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.ameya</groupId>
  <artifactId>012-TDD</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>012-TDD</name>
  <description>Test Driven Development - Testing services</description>
  <organization>
    <name>Vinsys</name>
    <url>https://services.vinsys.com</url>
  </organization>
  <inceptionYear>2021</inceptionYear>
  <distributionManagement>
    <repository>
        <url>https://mvnrepository.com/</url>
        <id>test</id>
    </repository>
  </distributionManagement>
  <properties>
    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
  </properties>
  <scm>
    <tag>dev</tag>
    <url>https://github.com/amsalways/training_material.git</url>
  </scm>
  <contributors>
    <contributor>
        <name>Avani</name>
        <email>avani@abc.com</email>
    </contributor>
  </contributors>
  <developers>
    <developer>
        <id>d1</id>
        <name>Ameya</name>
        <email>ameya.joshi@vinsys.com</email>
```

```
    </developer>
  </developers>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/junit/junit -->
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
        <!-- https://mvnrepository.com/artifact/org.mockito/mockito-all
        -->
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>2.28.2</version>
        <scope>test</scope>
    </dependency>
    <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-simple</artifactId>
            <version>1.7.30</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>1.7.30</version>
        </dependency>
  </dependencies>
  <reporting>
  <plugins>
        <plugin>
<!-- Generate the Report as Part of Project Reports ...
To generate the jacoco report as part of the site generation, add the
following in the <reporting> -->
            <groupId>org.jacoco</groupId>
            <artifactId>jacoco-maven-plugin</artifactId>
            <reportSets>
                <reportSet>
                    <reports>
                        <report>report</report>
```

```
                    </reports>
                </reportSet>
            </reportSets>
        </plugin>
    <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-report-plugin</artifactId>
            <version>3.0.0-M5</version>
        </plugin>
  </plugins>
  </reporting>
  <build>

    <plugins>
        <plugin>
            <groupId>org.jacoco</groupId>
            <artifactId>jacoco-maven-plugin</artifactId>
            <version>0.8.7</version>
            <executions>
                <execution>
                    <goals>
<!-- Prepares a property pointing to the JaCoCo runtime agent that can
be
passed as a VM argument to the application under test
-->
                        <goal>prepare-agent</goal>
                    </goals>
                </execution>
                <execution>
                    <id>report</id>
<!-- perform any operations necessary to prepare a package before
the actual packaging. This often results in an unpacked, processed
version of the package. -->
                    <phase>prepare-package</phase>
                    <goals>
<!-- It creates code coverage reports from the execution data
recorded by
 the JaCoCo runtime agent. -->
                        <goal>report</goal>
                    </goals>
                </execution>
```

```
                </executions>
            </plugin>
               <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-site-plugin</artifactId>
                <version>3.9.1</version>
            </plugin>
             <plugin>
                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-project-info-reports-plugin</artifactId
                >
                <version>3.1.2</version>
            </plugin>
        <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.7.0</version>
                <configuration>
                    <release>11</release>
                </configuration>
            </plugin>
    </plugins>
  </build>
</project>
--------------MyServiceTest.java-------------
package com.ameya.tests;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.fail;
import static org.mockito.Mockito.when;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import org.mockito.Mockito;

import com.ameya.daos.OrderDao;
import com.ameya.services.OrderService;
```

```
154 import com.ameya.services.OrderServiceImpl;
155
156 public class MyServiceTest {
157
158     OrderDao dao;
159     OrderService service;
160     int arr1[]=null;
161     int arr2[]=null;
162     int arr3[]=null;
163
164     @BeforeClass
165     public static void setUpOnce() {
166         System.out.println("This is set up before class");
167     }
168     @AfterClass
169     public static void tearDownClass() {
170         System.out.println("This is tear down after class");
171     }
172     @Before
173     public void setUpTest() {
174         arr1= new int[3];
175         arr1[0]=24;arr1[1]=15;arr1[2]=3;
176         arr2=new int[1];
177         arr2[0]=15;
178         arr3=new int[3];
179         arr3[0]=-3;arr3[1]=-2;arr3[2]=-5;
180         dao=Mockito.mock(OrderDao.class);
181         service=new OrderServiceImpl(dao);
182         System.out.println("++++ Test Data Set Up ++++");
183     }
184
185     @After
186     public void tearDownTest() {
187         service=null;
188         arr1=null;
189         arr2=null;
190         arr3=null;
191         System.out.println("++++ Test Data Cleaned Up ++++");
192     }
193
194     @Test
```

```
195     public void testFindTheGreatestFromAllData(){
196         when(dao.getAllData()).thenReturn(arr1);
197         assertEquals(24,service.findTheGreatestFromAllData());
198         //assertEquals(24,OrderService.test()); -->static methods also
            can be tested
199     }
200     @Test
201     public void testFindTheGreatestFromAllData_ForOneValue() {
202         when(dao.getAllData()).thenReturn(arr2);
203         assertEquals(15,service.findTheGreatestFromAllData());
204     }
205     @Test
206     public void testFindTheGreatestFromAllData_NoValues() {
207         when(dao.getAllData()).thenReturn(new int[] {});
208         assertEquals(0,service.findTheGreatestFromAllData());
209     }
210     @Test
211     public void testFindTheGreatestFromAllData_NegativeValues() {
212         when(dao.getAllData()).thenReturn(arr3);
213         assertEquals(-2,service.findTheGreatestFromAllData());
214     }
215 }
216 ------------OrderDao.java-------------
217 package com.ameya.daos;
218
219 public interface OrderDao {
220
221     int [] getAllData();
222 }
223 ------------OrderService.java-------------
224 package com.ameya.services;
225
226 public interface OrderService {
227     int findTheGreatestFromAllData();
228      static void test() {
229          System.out.println("Test");
230      }
231 }
232 -----------OrderServiceImpl.java-------------
233 package com.ameya.services;
234
```

```
import com.ameya.daos.OrderDao;

public class OrderServiceImpl implements OrderService {

    private OrderDao dao;
    public OrderServiceImpl(OrderDao dao) {
        this.dao=dao;
    }
static void test() {
    System.out.println("Test");
}
    @Override
    public int findTheGreatestFromAllData() {
        int [] data=dao.getAllData();
        int greatest=Integer.MIN_VALUE;
        if(data.length==0) {
            return 0;
        }
        for(int value : data) {
            if(value>greatest) {
                greatest=value;
            }
        }
        return greatest;
    }
-------------MyDaoTests.java-------------
/*
 * package com.ameya.tests;
 *
 * import static org.junit.Assert.fail;
 *
 * import org.junit.Test;
 *
 * public class MyDaoTests {
 *
 * @Test public void testOne() { fail("Not Implemented Yet"); }
 *
 * @Test public void testTwo() { fail("Not Implemented Yet"); }
 *
 * @Test public void testThree() { fail("Not Implemented Yet"); } }
 */
```

```
276   ------------ApplicationTestSuite.java-------------
277   /*
278   * package com.ameya.tests;
279   *
280   * import org.junit.runner.RunWith; import org.junit.runners.Suite;
      import
281   * org.junit.runners.Suite.SuiteClasses;
282   *
283   * @RunWith(Suite.class)
284   *
285   * @SuiteClasses({ MyServiceTest.class, MyDaoTests.class }) public class
286   * ApplicationTestsSuite {
287   *
288   * }
289   */
290
291 }
292
```