**Exercise 3: Test Execution Order**

**Scenario:** You want to control the order in which tests are executed.

**OrderedTests.java :-**

package org.example;  
  
import org.junit.jupiter.api.\*;  
  
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)  
public class OrderedTests {  
  
 @Test  
 @Order(3)  
 void testDelete() {  
 System.*out*.println("Running testDelete()");  
 Assertions.*assertTrue*(true);  
 }  
  
 @Test  
 @Order(1)  
 void testCreate() {  
 System.*out*.println("Running testCreate()");  
 Assertions.*assertTrue*(true);  
 }  
  
 @Test  
 @Order(2)  
 void testUpdate() {  
 System.*out*.println("Running testUpdate()");  
 Assertions.*assertTrue*(true);  
 }  
}

**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  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>OrderedTestsDemo</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <!-- JUnit 5 Jupiter API -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- JUnit 5 Execution -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output :-**

