**Parul Sharma-5804 JVM Trainee**

Introduction to Dependency Management using Maven

Following parameters are to be considered for exercise submission:

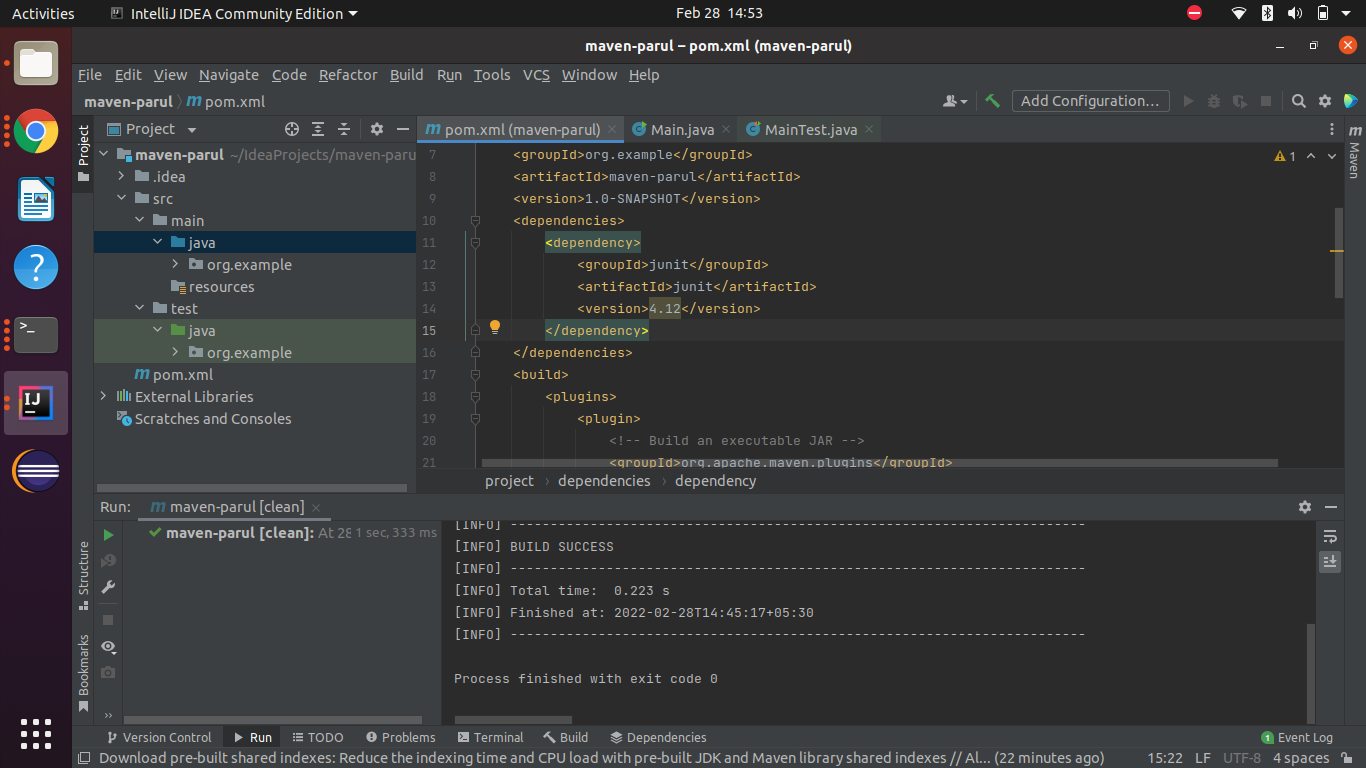
1. The complete project has to be submitted.
2. .idea, target directories should be excluded
3. A text file containing the relevant URLs which are referenced and any other documentation.

Github Project url : https://github.com/parulsharma01/JavaPrograms/tree/MavenExcercise

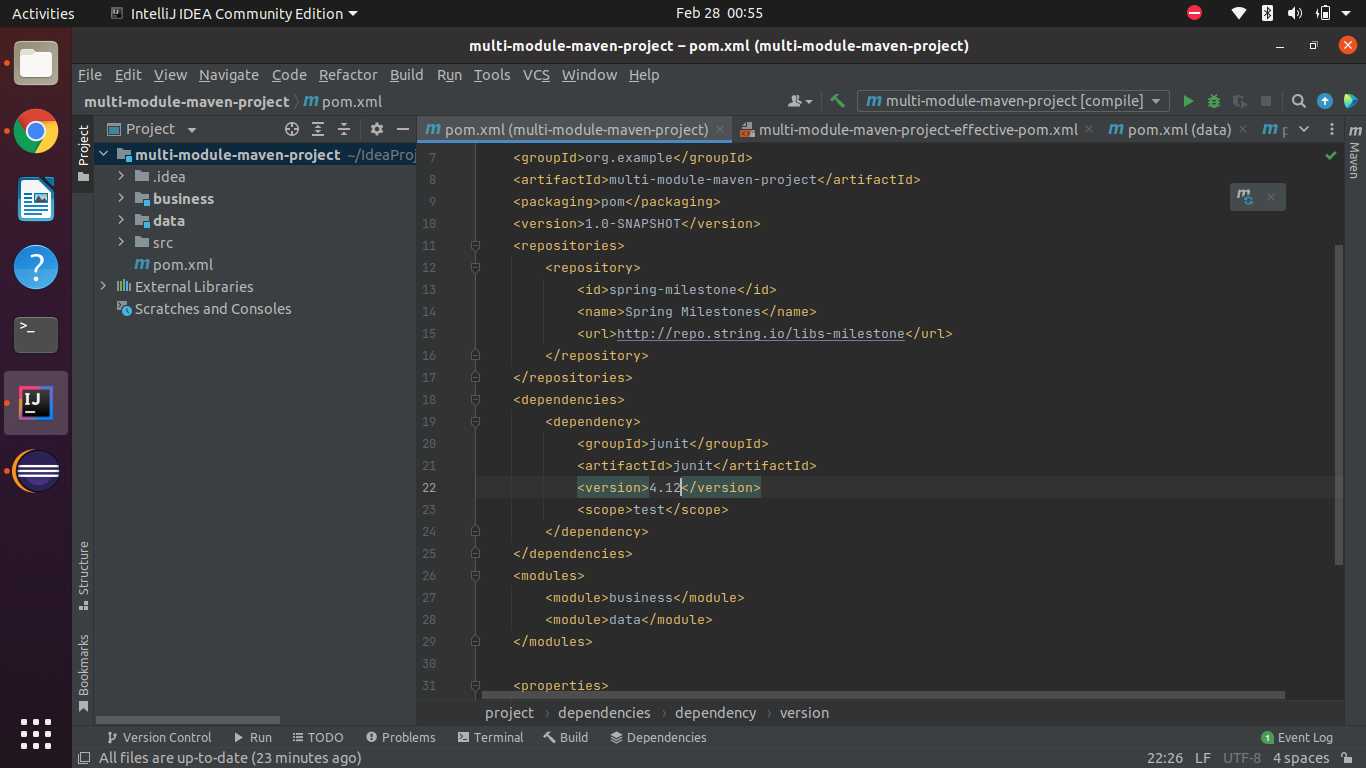
Questions:

1. Add a maven dependency and its related repository URL.

Repo URL : **https://mvnrepository.com/artifact/junit/junit/4.12**



1. Add a new repository in the pom.xml and use its dependencies.



1. Using JAR plugin, make changes in the pom.xml to make the jar executable. Using java -jar JAR\_NAME, the output should be printed as "Hello World"

We’ll perform the commands

Maven>project-name>Lifecycle

1.clean

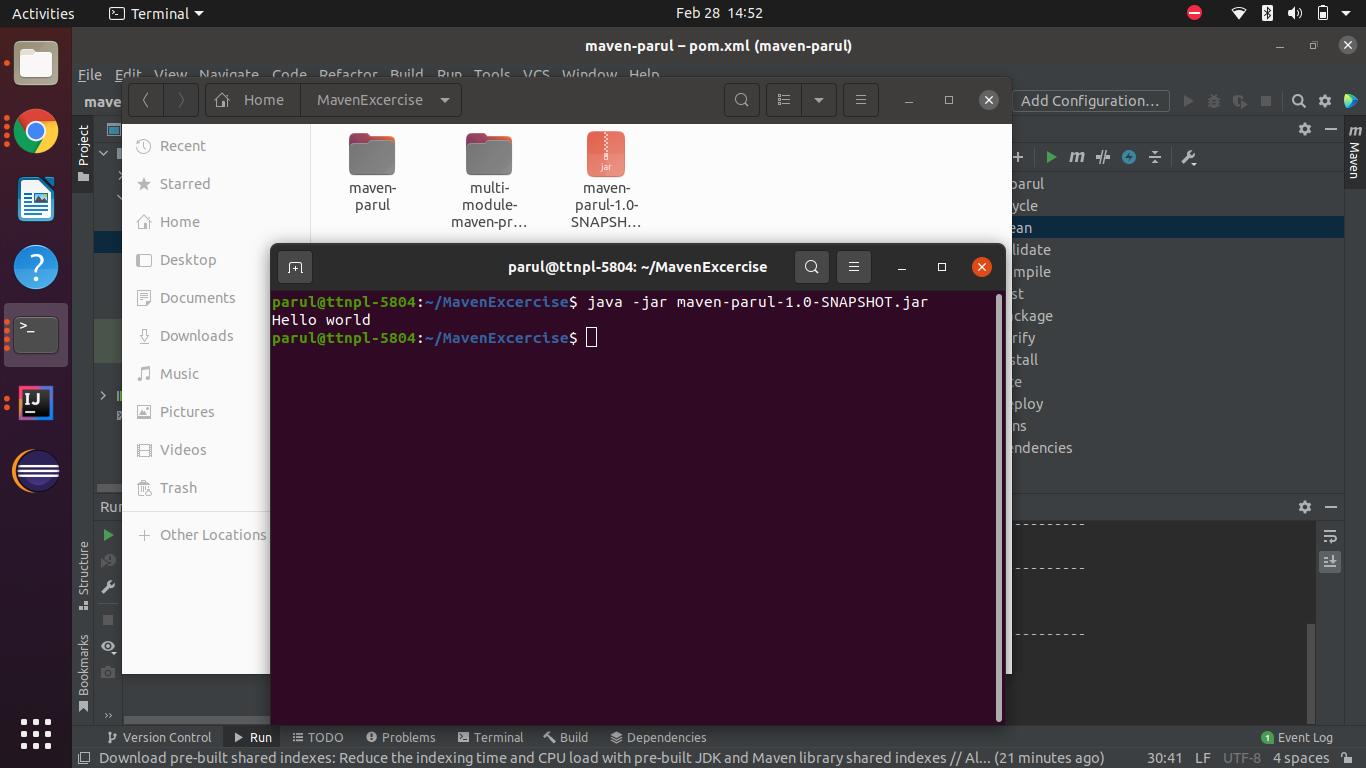
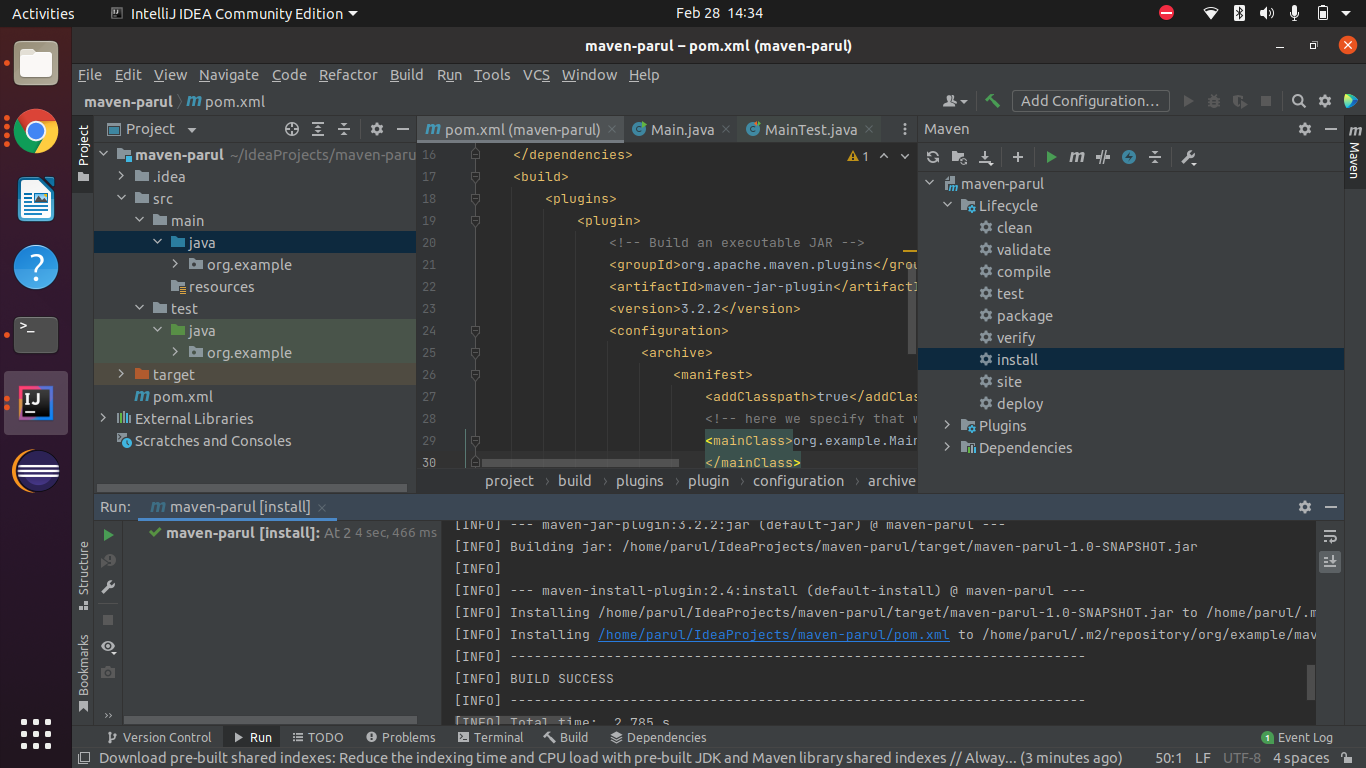
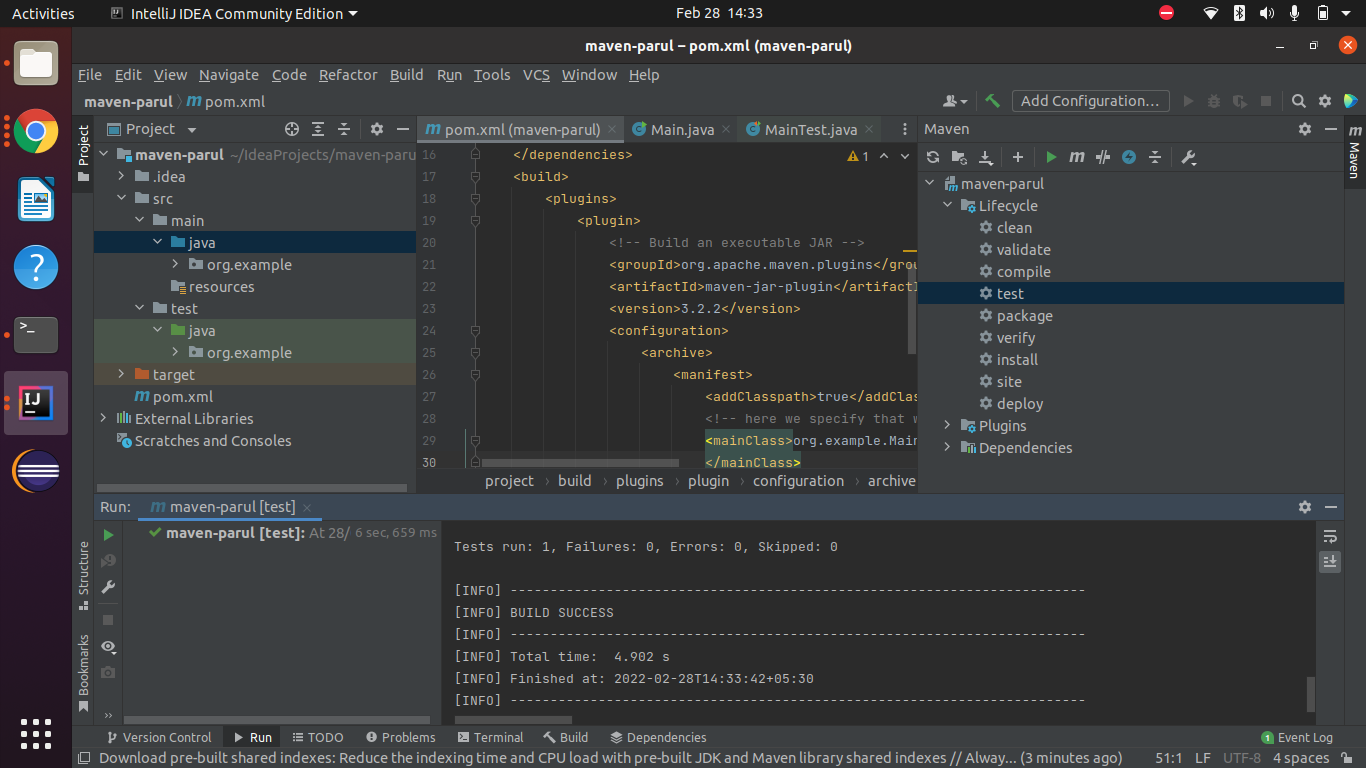
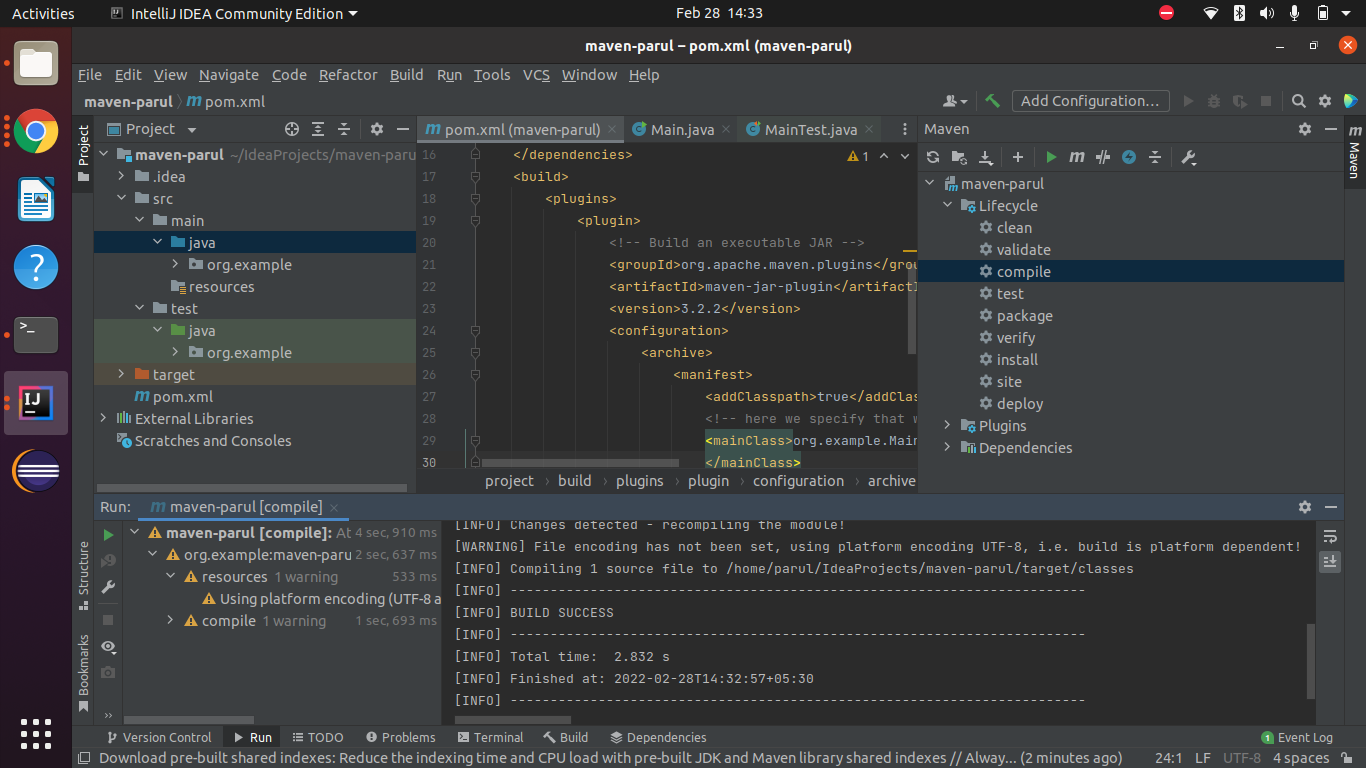
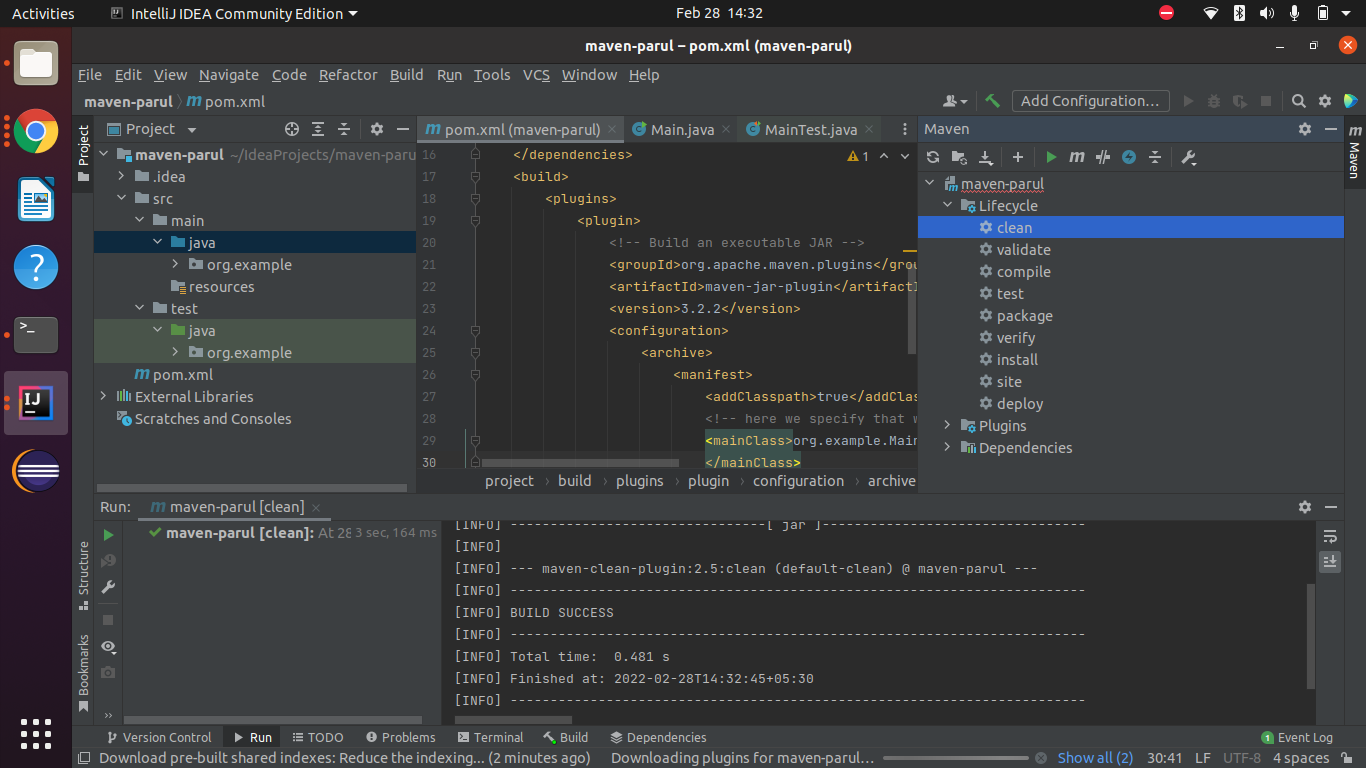
2.compile

3.test

4.install

We will get .jar file in the target directory.

Please refer to the maven-parul-1.0-SNAPSHOT.jar



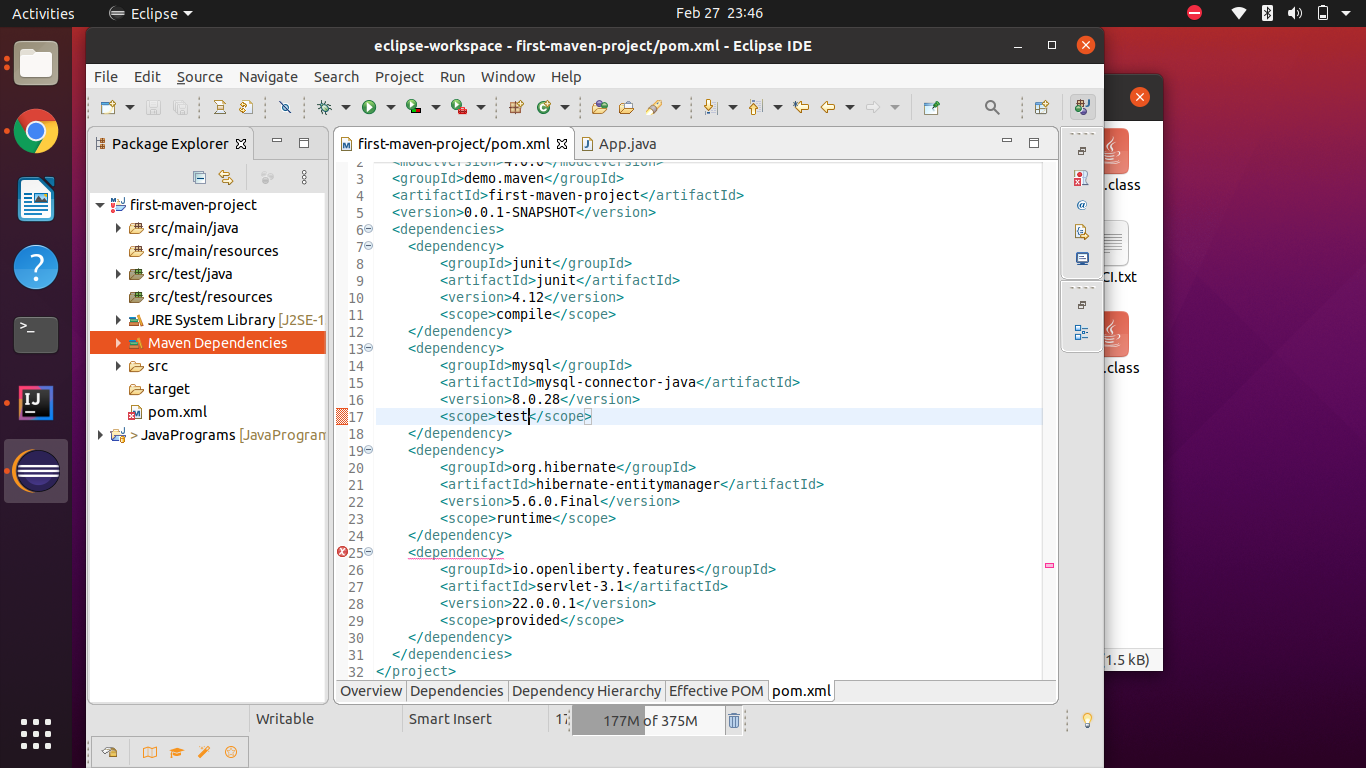
1. Differentiate between the different dependency scopes: compile, runtime, test, provided using different dependencies being defined in your pom.xml.

compile : available while compiling, runtime, test, and in jar file also

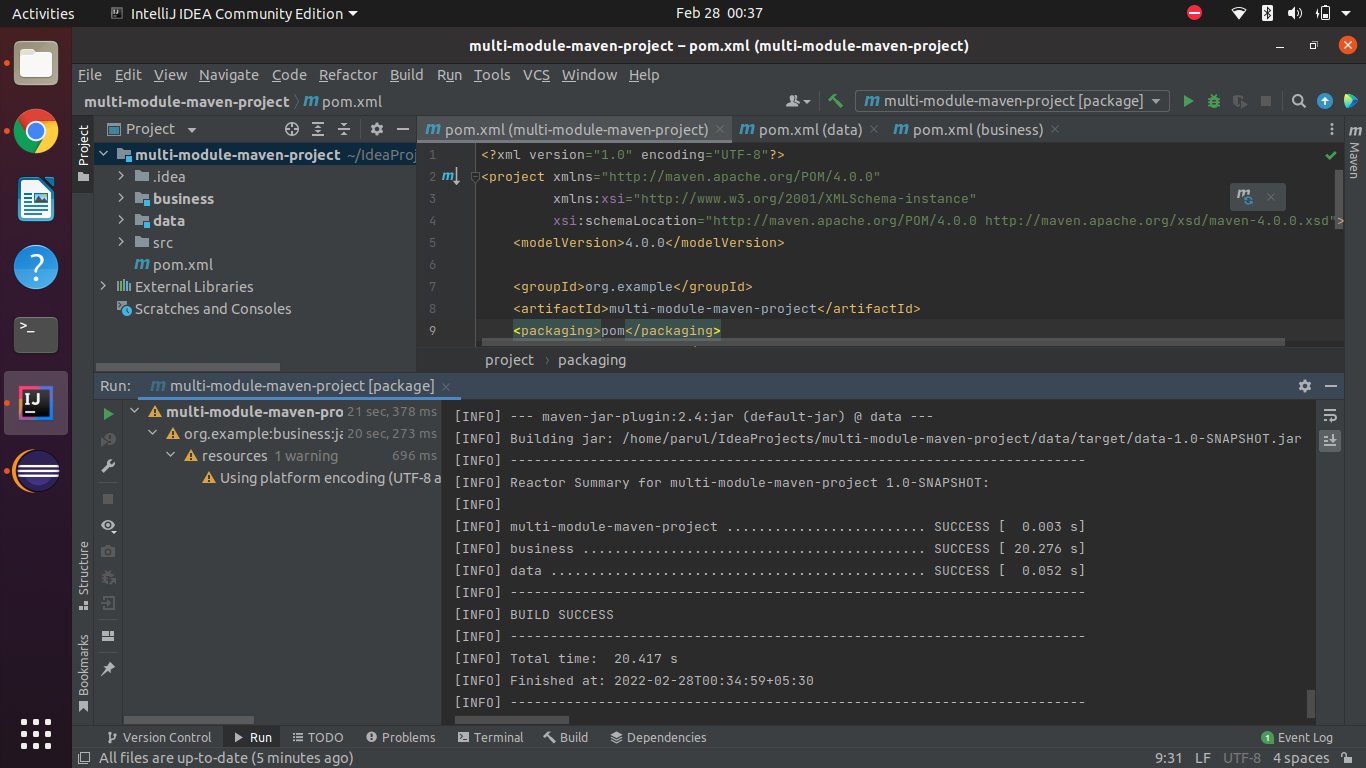
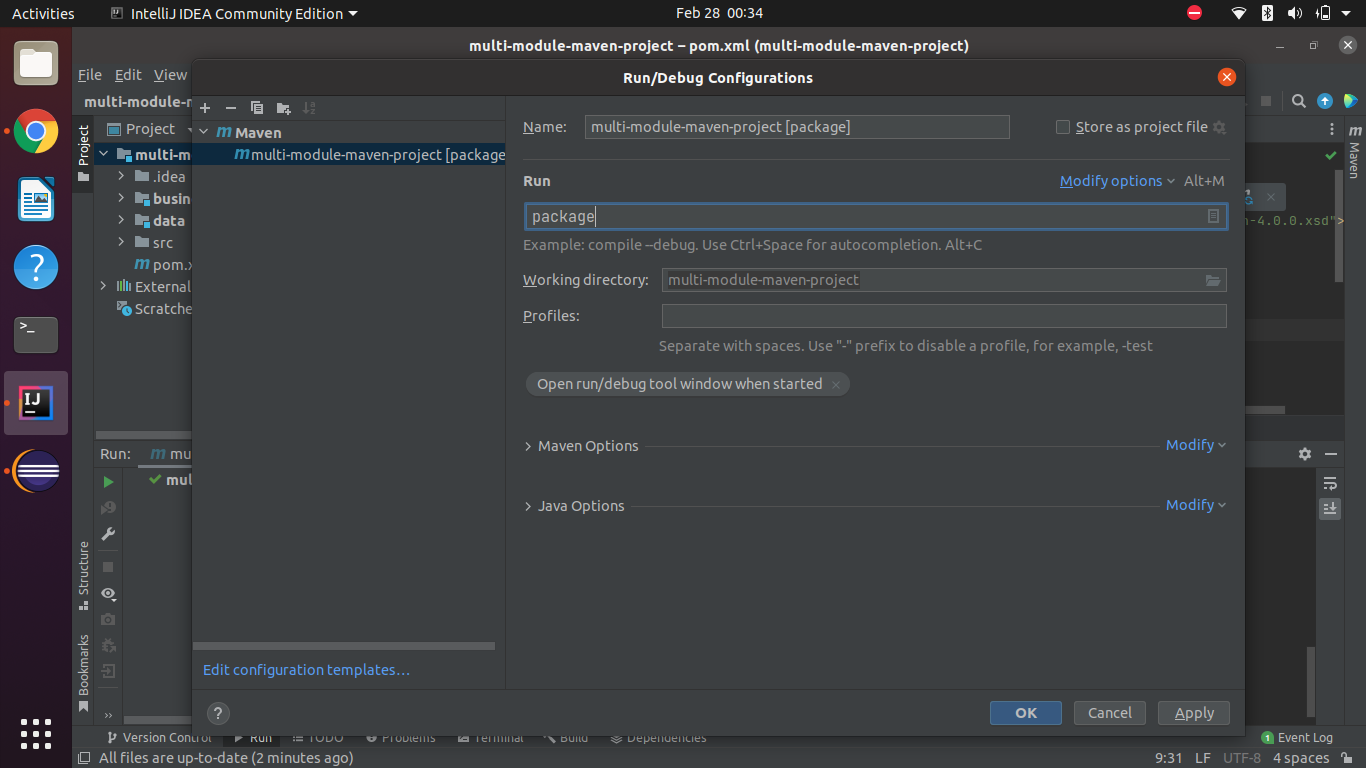
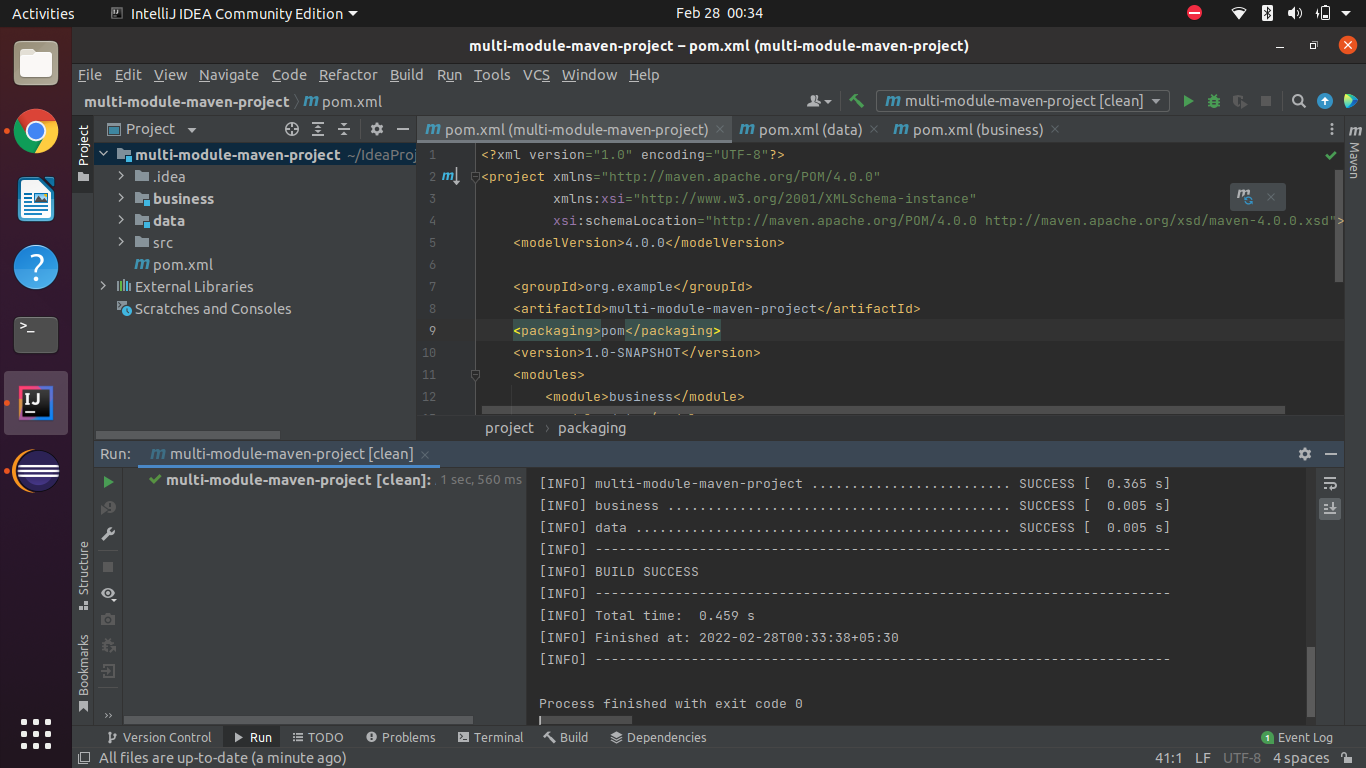
test : only available for test, it is only available for the code which is inside the source test java folder.

runtime : not available in compile time, available in runtime and in jar file.

provided : available in compile, test and runtime but not included in jar file.



1. Create a multi-module project. Run package command at the top level to make jar of every module.



Project Github URL :

https://github.com/parulsharma01/JavaPrograms/tree/MavenExcercise/multi-module-maven-project

**Class Notes :**

* Exclusions of dependencies :

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>com.oracle.oci.sdk</groupId>

<artifactId>oci-java-sdk-common</artifactId>

</exclusion>

</exclusions>

</dependency>

* Range of version : [ ] closed brackets are used and it means both values are included.

for eg.

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>[4, 8.0]</version>

<scope>test</scope>

</dependency>

[4, 8.0] : include any jar of min version 4 and max version of 8.0

[4, 8) : jar can be 4 to version less than 8(not included)

[4, ] : min version is 4 and max version is latest version

[, 8] : any version below 8.