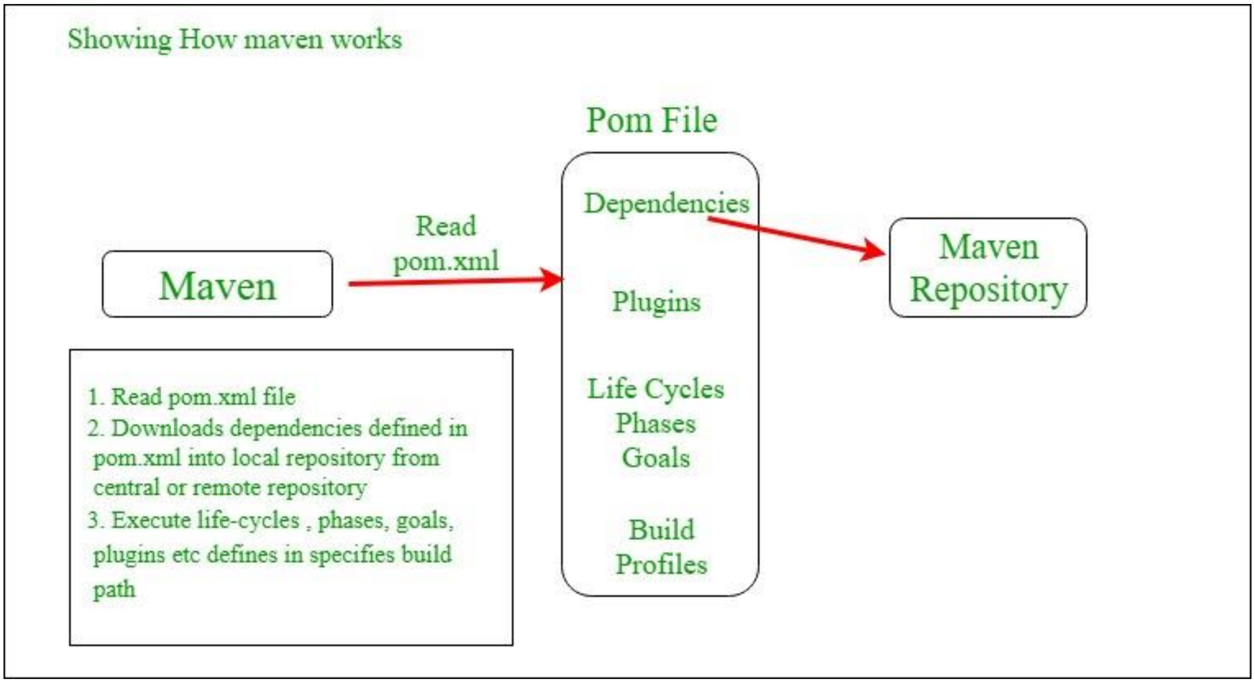
Contents

- Maven (Need , POM.xml, Maven Repository and Types, Maven Lifecycle)

MAVEN : Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. But it is too much advanced than ANT.

In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.


Showing How maven works
Pom File
Maven
Read pom.xml
Dependencies
Plugins
Life Cycles
Phases
Goals
Build
Profiles
Maven Repository
1. Read pom.xml file
2. Downloads dependencies defined in pom.xml into local repository from central or remote repository
3. Execute life-cycles , phases, goals, plugins etc defines in specifies build path

Let us have a brief look at the very informative article :

https://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-java-projects/

What are the shortcomings of not using Maven ?

Challenge 1 : Lets create a java project without maven and try to execute a create table query using jdbc protocol . Suppose you a database with the name test_db_jbdl . What would happen if you run this code ?

```
 public static void main(String[] args) throws SQLException {

        Connection connection = DriverManager.getConnection(

                "jdbc:mysql://127.0.0.1:3306/test_db_jbdl", "root", "");



        Statement statement = connection.createStatement();

        statement.execute("create table maven_dummy(id int, name varchar(30), count
int)");

}
```

Will there be any (i) compilation error (ii) runtime error ?

HINT : Click on the execute method of statement object for its implementation .

Challenge 2 : How to manage dependencies without Maven ?

HINT : We can download the dependency as a jar and deploy it in our project structure .

File > ProjectStructure > Library > Add New > Java

HOW TO ADD DEPENDENCIES USING MAVEN

Challenge 1 : What is groupId , ArtifactId and version in a maven project?

HINT : https://maven.apache.org/guides/mini/guide-naming-conventions.html

groupId : unique entity or organization

artifactId : unique project

Challenge 2 : Create a maven project and add the following dependencies using spring initialiser :

-Lombok

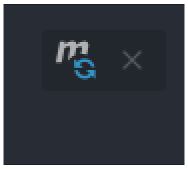
Now create a Person class with id , name and age as properties . Create a person object using lombok dependency to test if the dependency works .

Challenge 3 : Maven adds any child dependencies for the dependency we need to add to our project . If we put the jar ourselves in the library path of the project , do the child dependencies if any get fulfilled ?

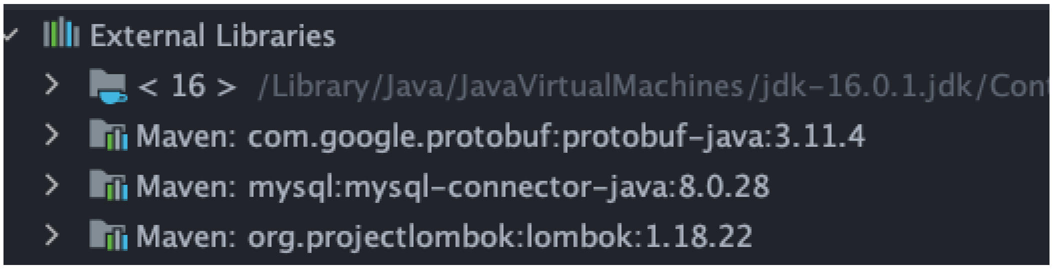
Challenge 4 : What is pom.xml and how to add dependencies in pom.xml ?

HINT : A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

Challenge 5 : Does the dependency get downloaded just by adding it to pom.xml ?

HINT :  Through this button , we have to load the changes and download the dependency from central repository . https://repo.maven.apache.org/maven2/

Challenge 6 : What is the difference b/w External Libraries of Java Project vs External Libraries of Maven Project ?



```
External Libraries
  > < 16 > /Library/Java/JavaVirtualMachines/jdk-16.0.1.jdk/Con
  > Maven: com.google.protobuf:protobuf-java:3.11.4
  > Maven: mysql:mysql-connector-java:8.0.28
  > Maven: org.projectlombok:lombok:1.18.22
```

HINT : groupId, artifactId and version together combine to make an archetype . Dependency of Maven project is also Maven project .

Challenge 7 : Where are parent/child dependencies of any dependency you add in pom.xml defined ?

HINT : In your project's pm.xml , click on artifactId of the dependency for which you need to know the parent/child dependencies .

Challenge 8 : Why do we need the parent dependencies for our dependency in pom.xml ? Is there really a need ?

Challenge 9 : What is the scope tag inside a dependency in pom.xml ?

HINT : https://stackoverflow.com/questions/26975818/what-is-scope-under-dependency-in-pom-xml-for

]

Challenge 10 : How to search for mysql-connector-java dependency in the Official Central repository ?

HINT : Go to https://repo.maven.apache.org/maven2/

First search for group id and then artifact id and then the (latest) version .

MAVEN REPOSITORY

Maven Repository

Maven repositories are directories of packaged JAR files with some metadata. The metadata are POM files related to the projects each packaged JAR file belongs to, including what external dependencies each packaged JAR has. This metadata enables Maven to download dependencies of your dependencies recursively until all dependencies are downloaded and put into your local machine.

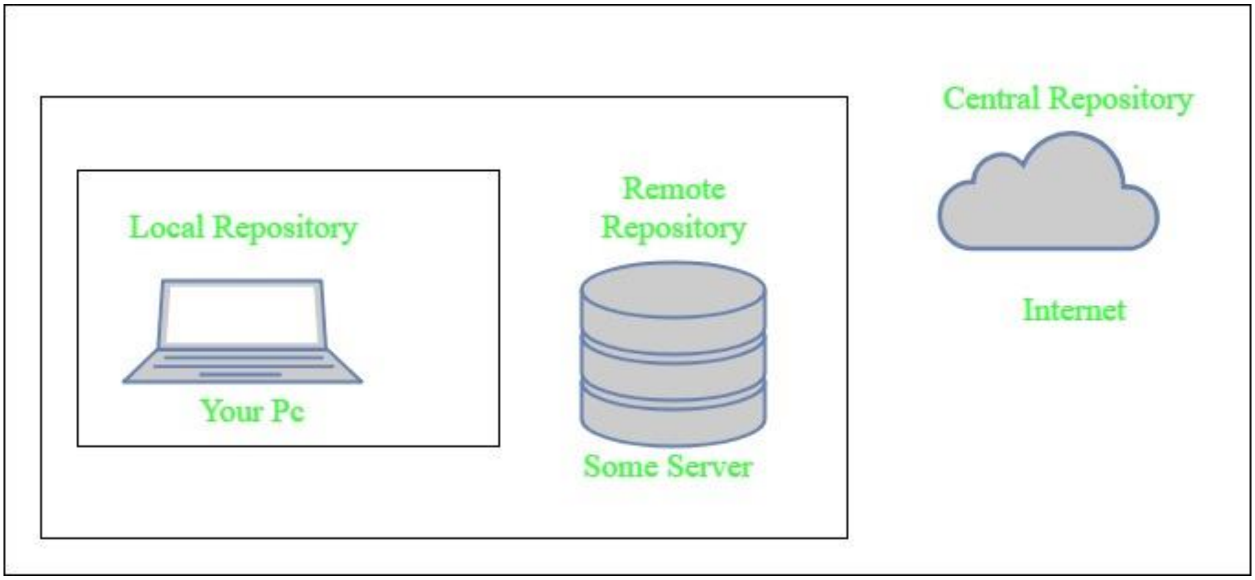
Maven has three types of repository :

1. Local repository (dependencies downloaded in your local computer)

2. Central repository (public dependencies available on net)

3. Remote repository (private to an organization available on internet)

Maven searches for dependencies in this repositories. First maven searches in Local repository then Central repository then Remote repository if Remote repository specified in the POM.



Local repository- A local repository is a directory on the machine of developer. This repository contains all the dependencies Maven downloads. Maven only needs to download the dependencies once, even if multiple projects depends on them (e.g. ODBC).

By default, maven local repository is user_home/m2 directory.

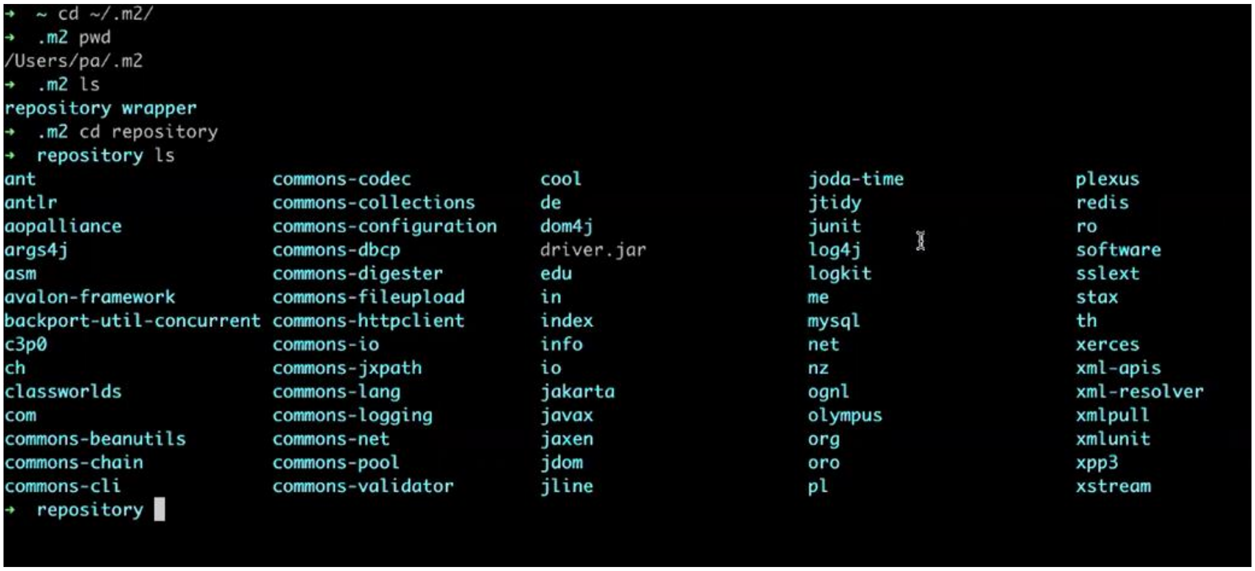
example – C:\Users\asingh\.m2

Central repository- The central Maven repository is created Maven community. Maven looks in this central repository for any dependencies needed but not found in your local repository. Maven then downloads these dependencies into your local repository.

Official Maven Central Repository ~ https://repo.maven.apache.org/maven2/

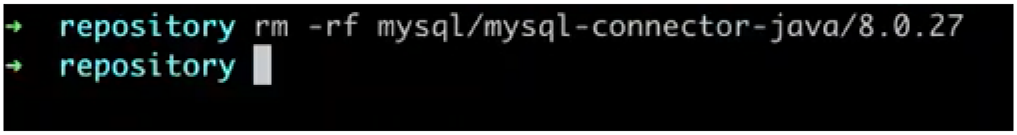
Remote repository- remote repository is a repository on a web server from which Maven can download dependencies.it often used for hosting projects internal to organization. Maven then downloads these dependencies into your local repository.

Challenge 1 : Lets explore the local repository in your .m2 folder of your computer



```
→  ~ cd ~/.m2/
→  .m2 pwd
/Users/pa/.m2
→  .m2 ls
repository wrapper
→  .m2 cd repository
→  repository ls
ant                      commons-codec            cool                     joda-time                plexus
antlr                    commons-collections      de                       jtidy                    redis
aopalliance              commons-configuration    dom4j                    junit                    ro
args4j                   commons-dbcp             driver.jar               log4j                    software
asm                      commons-digester         edu                      logkit                   sslext
avalon-framework         commons-fileupload       in                       me                       stax
backport-util-concurrent commons-httpclient       index                    mysql                    th
c3p0                     commons-io               info                     net                      xerces
ch                       commons-jxpath           io                       nz                       xml-apis
classworlds              commons-lang             jakarta                  ognl                     xml-resolver
com                      commons-logging          javax                    olympus                  xmlpull
commons-beanutils        commons-net              jaxen                    org                      xmlunit
commons-chain            commons-pool             jdom                     oro                      xpp3
commons-cli              commons-validator        jline                    pl                       xstream
→  repository 
```

Challenge 2 : What will happen if I delete a dependency version which my maven project is currently using ?



```
→  repository rm -rf mysql/mysql-connector-java/8.0.27
→  repository
```

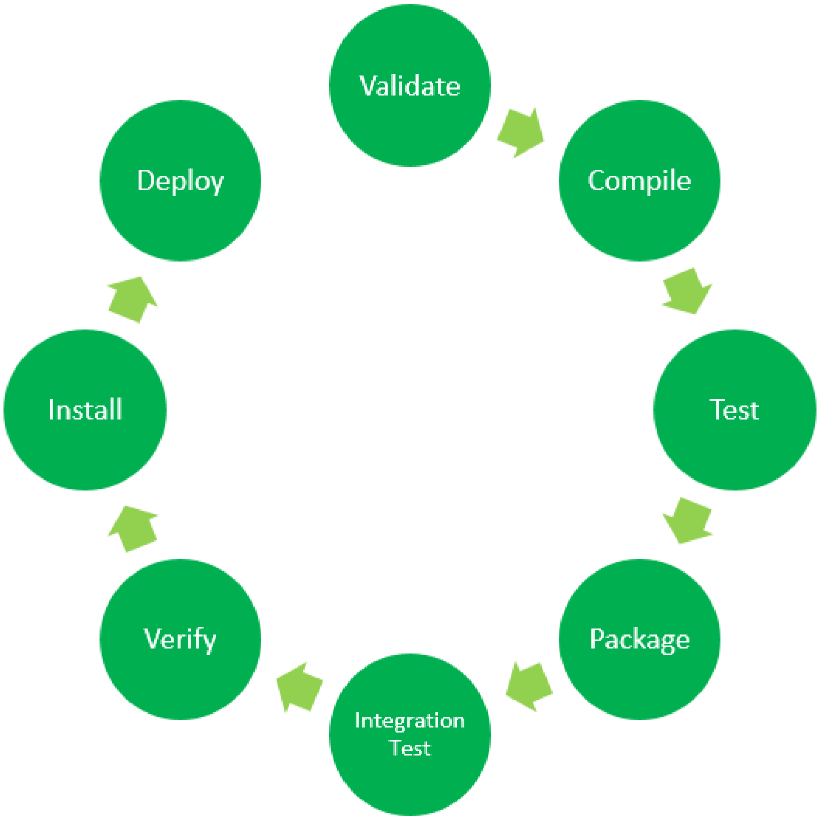
HINT : It does not give any error immediately but we see dependencies is not there in External Libraries and when we run the java class , it throws SQL exception : No suitable driver found .

Challenge 3 : If we change the directory inside the terminal and go to our maven project and then run the `mvn package`, what will happen ?

HINT : We need to understand the Maven Build Lifecycle for this .

- `validate` - validate the project is correct and all necessary information is available
- `compile` - compile the source code of the project
- `test` - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- `package` - take the compiled code and package it in its distributable format, such as a JAR.
- `verify` - run any checks on results of integration tests to ensure quality criteria are met
- `install` - install the package into the local repository, for use as a dependency in other projects locally

- deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.



Maven Commands

mvn clean: Cleans the project and removes all files generated by the previous build.

mvn validate : (intermediary phase)Validates the src folder to check if code files are placed correctly .

mvn compile: Compiles source code of the project into target

mvn test-compile: Compiles the test source code.

mvn test: Runs tests for the project.

mvn package: compiles , runs tests and Creates JAR or WAR file for the project to convert it into a distributable format.

mvn verify: checks if the jar is created .

mvn install: Deploys the packaged JAR/ WAR file to the local repository.

mvn deploy: Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project.

(Contd) Challenge 3 : We deleted the dependency using the terminal from local repository . If we change the directory inside the terminal and go to our maven project and then run the mvn package, what will happen ?

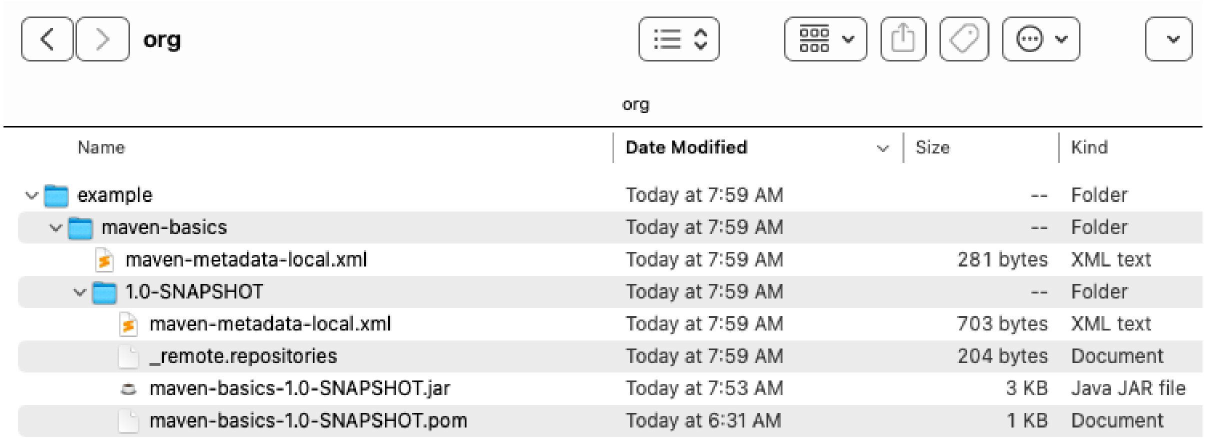
HINT : It will download from central .

Challenge 4 : Now if we run the maven package once again , will it again download from the central repository or refer to the dependency in the local repository (.m2 folder) ?

Challenge 5 : Does mvn package or mvn install also perform mvn clean ?

HINT : Generally when we run any of the above commands, we add the mvn clean step so that the target folder generated from the previous build is removed before running a newer build. This is how the command would look on integrating the clean step with install phase: mvn clean install .

Challenge 6 : What is the difference b/w mvn package vs mvn install ?



Name	Date Modified	Size	Kind
example	Today at 7:59 AM	--	Folder
maven-basics	Today at 7:59 AM	--	Folder
maven-metadata-local.xml	Today at 7:59 AM	281 bytes	XML text
1.0-SNAPSHOT	Today at 7:59 AM	--	Folder
maven-metadata-local.xml	Today at 7:59 AM	703 bytes	XML text
_remote.repositories	Today at 7:59 AM	204 bytes	Document
maven-basics-1.0-SNAPSHOT.jar	Today at 7:53 AM	3 KB	Java JAR file
maven-basics-1.0-SNAPSHOT.pom	Today at 6:31 AM	1 KB	Document

HINT : mvn install also installs your current project in your .m2/repository folder so other project can use it as an archetype or dependency
https://stackoverflow.com/questions/16602017/how-are-mvn-clean-package-and-mvn-clean-install-different

Challenge 7 : If we have no compile time errors in classes inside src/main/java folder but we have a compile time error in a class inside src/test/java folder .

(i) Now we run mvn compile , will we get a compile time error or our src/main/java code will be successfully compiled and put into the target folder ?

(ii) Will we have the jar file created inside the target folder ?

(iii) Will we get an error if I run the mvn test instead of mvn compile ?

Challenge 8 : If we have compile time error in src/main/java folder and no compile time error in src/test/java folder . What will happen if I run the mvn test ?

HINT : maven lifecycle phases are sequential . If you run the mvn test then mvn validate ,mvn compile will also run .

Challenge 9 : Create a new maven project called Calculator . Inside src/main/java , create a class Calcy with function definitions for add , multiply , divide and subtraction of two numbers .

Now , your task is to be able to use this Calculator maven project as a dependency in your previous maven project and find out what is 4 times 80 ?

HINT : mvn install

Challenge 10 : Now if I create another maven project named Business and include the dependency of our original maven project in its pom , will the Business project also contain the dependency of the Calculator project ? (Given that we have run command : mvn install for Original maven basics project)

Challenge 11 : What if I delete the Calculator dependency from local repository , then :

(i) If there is code using Calci class in Business project or Original project class files , will they throw a compile time error ?

(ii) How to remediate it ?

HINT : run mvn install command for Calculator project to make it install in local repository .

Challenge 12 : If I go to my Calculator project and add a new method of calculatePower inside the Calci class . Our task is to find 3^4 by calling Calci.power(3,4) inside the Business Project

(i) Now will we able to call this new function inside the Business project ?

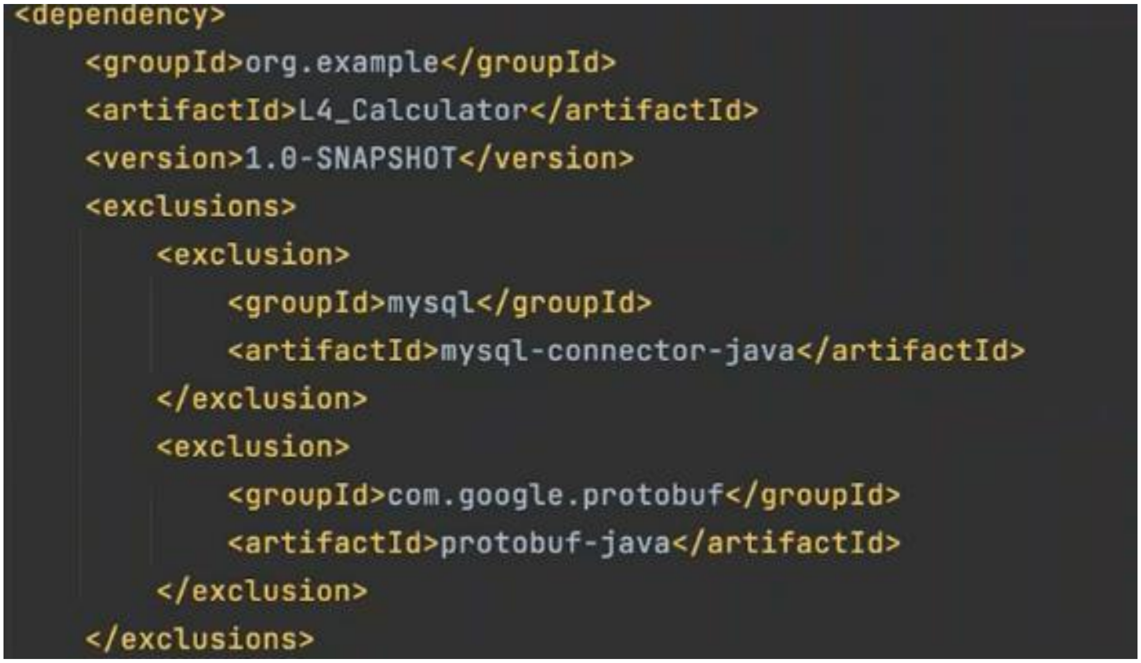
(ii) If I do mvn package , now I can use this function insdie Business project ?

(iii) If I do mvn install , now I can use this function insdie Business project ?

Challenge 13 : What is the difference between mvn package and mvn clean package ?

Challenge 14 : If I had a maven project A with dependency mysql-connector-java v10 and another maven project B with dependency mysql-connector-java v12 . Now in my maven project C , I have project A and B as parent dependencies . Which version of mysql-connector-java will be present in the external libraries of Project C ?

HINT : position of defining dependencies A and B in the pom . In case of the same dependencies , version will be of whichever is first in pom of C . If we don't want to rely on position in POM , then we can use exclusions .If we want to use the version of mysql-connector-java from Project A , then we can use exclusion tag while defining the dependency of B in pom of C .



```
<dependency>
    <groupId>org.example</groupId>
    <artifactId>L4_Calculator</artifactId>
    <version>1.0-SNAPSHOT</version>
    <exclusions>
        <exclusion>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </exclusion>
        <exclusion>
            <groupId>com.google.protobuf</groupId>
            <artifactId>protobuf-java</artifactId>
        </exclusion>
    </exclusions>
```

Challenge 15 : How to rename the target folder to new?

HINT : In the pom.xml , add the following post your dependencies tag :

<build>

<directory>/Users/aadhar/Desktop/JBDL-master/L4_maven-basics/new</directory>

</build>

Challenge 16 : (HomeWork) How to change the path of the .m2 folder using pom.xml ?

References :

https://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-java-projects/ IMP

Official Maven Documentation by Apache : https://maven.apache.org/

https://repo.maven.apache.org/maven2/ Official Maven Central Repository

https://mvnrepository.com/ Unofficial Maven Central Resource

https://stackoverflow.com/questions/39185798/what-is-the-difference-between-artifactid-and-groupid-in-pom-xml IMP

https://stackoverflow.com/questions/14725316/what-is-the-use-of-pom-xml-in-maven

https://maven.apache.org/guides/introduction/introduction-to-repositories.html

https://www.geeksforgeeks.org/maven-lifecycle-and-basic-maven-commands/ IMP

https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html

https://stackoverflow.com/questions/6028534/how-to-exclude-dependency-in-a-maven-plugin IMP Exclusions

https://maven.apache.org/guides/introduction/introduction-to-optional-and-excludes-dependencies.html IMP Exclusions

LIVE
Java
Backend
Development