***MAVEN:***

Maven is a tool used to simplify the build process by managing dependencies, compilation of the code and package management. Maven uses POM.xml file for reference. Project Object Model (POM) holds the project information, Dependencies , Configuration & Plugins which are required for that project.

We need to install maven from maven/apache.

After installing the maven, we can check maven version using ***mvn –version*** command on our cmd.

We can create maven project using command line through Interactive and non-interactive mode.

Non-Interactive mode:

***mvn archetype:generate -DgroupId=com.epam -DartifactId=DemoMavenProject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false***

Interactive mode: we need to run commands as follows:

1. mvn archetype:generate
2. select the artifact type
3. select version
4. enter groupId
5. enter artifactId
6. enter Snapshot version
7. enter package name.

*After creating the project and launching it in Eclipse IDE. I got to know about the project structure which includes source folder which is src/main/java, target folder, POM.xml file, src/test/java and the resources folders src/main/resources which were missing or hidden so either we need to create them or we need to configure them.*

I found out that there are three major built-in **Build Life Cycles**:

1. default
2. clean
3. site

Each Build Lifecycle is made up of Phases:

***Default Lifecycle phases:***

* ***Validate:*** After running the mvn validate phase I found out that it is used to check the information about the project, to check whether all the information are correct or not.

A screenshot of a computer

Description automatically generated

* ***Compile:*** I got to know that it will take all the java code which are in src/main/java and it will compile that code and will generate the byte code and few folders are created.
* ***Folders created:***

target/classes

classes files

plugins got downloaded.

A screenshot of a computer

Description automatically generated

* ***Test:*** it will run all the test classes related to src/main/test.

A computer screen shot of a program

Description automatically generated

* ***Package:*** I found that it take compiled java code and package it in distributed format JAR

Which will be present inside the target folder.

* ***Install:*** It will install package in local repository as dependency to use for the other projects.

A screenshot of a computer program

Description automatically generated

Run java class from through maven in cmd.

A screenshot of a computer

Description automatically generated