**3. Maven**

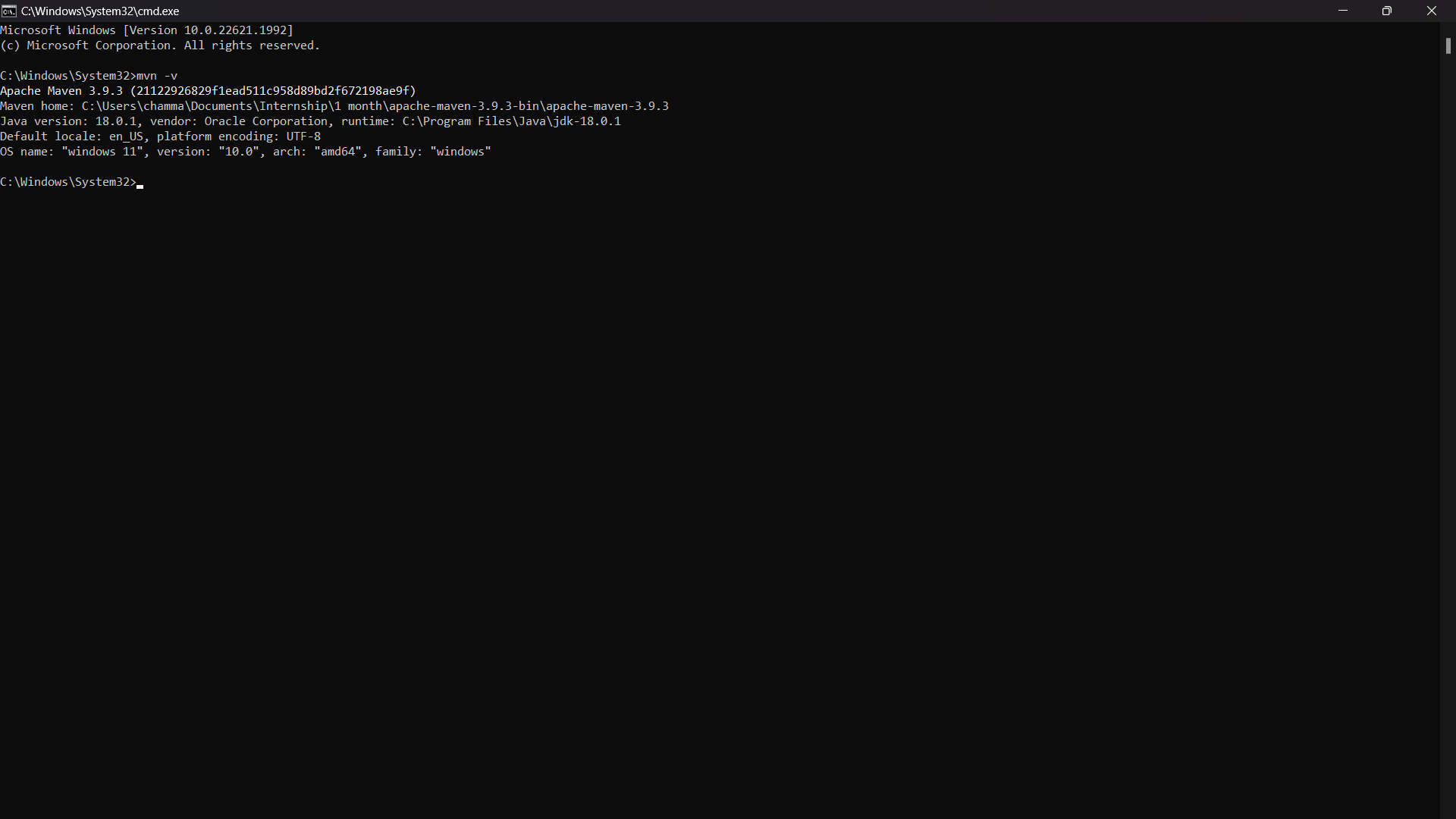
1. Why do we need a build tool like maven?

A build tool like Maven is essential for software development projects for several reasons

1. Dependency Management
2. Consistent Build Process
3. Project Structuring
4. Automated Build Tasks
5. Plugin Ecosystem
6. Reproducibility
7. Integration with Continuous Integration
8. Compatibility with Various Languages

Maven simplifies the build process, promotes best practices, and streamlines the development workflow, making it an indispensable tool for efficient and scalable software development projects.

1. Install maven
2. Display output of maven version



1. What is the pom.xml file?

The pom.xml file, also known as the Project Object Model, is an essential configuration file used in Apache Maven-based projects. It defines the metadata and configuration for a Maven project, including its dependencies, build settings, plugins, and other project-related information.

1. Explain these tags found in pom.xml files?

|  |  |
| --- | --- |
| groupId | Represents the organization or group that created the project. |
| artifactId | The project's unique identifier |
| version | Specifies the project's version number. |
| packaging | Specifies the type of artifact that the project produces. |
| dependencies | Section lists all the external libraries and frameworks (dependencies) that the project relies on. |
| dependency | Specifying its groupId, artifactId, and version. |
| properties | Allows defining project-specific properties that can be referenced throughout the pom.xml. It helps in centralizing and reusing configuration values. |

1. Create a method which accept an integer as parameter and returns the square of it
2. Add junit (v5) dependency
3. Add a unit test to test the method
4. Run unit tests with maven. What is the command you used?
5. Create a Student class with following attributes and add getters/setters

* id: int
* name: String
* age: int
* subjects: List<String>

1. In your main method create and student object and set these values

id: 1

name: “john”

age: 20

subjects: [“Maths”, “English”, “History”]

1. Then print these student values using getters (e.g: s.getName())
2. Add lombok dependency and remove getter/setter methods from Student class
3. Explain the usage of these commands

|  |  |
| --- | --- |
| mvn clean | Clean the project by deleting any generated files or directories created during the build process. It is often executed before starting a new build to ensure a clean and consistent build environment. The primary purpose of this command is to remove all the output generated from previous builds, including compiled classes, JARs, test results, and other build artifacts. |
| mvn install | Build the project, compile the source code, execute tests, and install the project's artifacts into the local Maven repository. It is a commonly used command, especially during development or when you want to share your project's artifacts with other local projects. |
| mvn package | Build the project and package the compiled code into an artifact without installing it into the local Maven repository. This is often used when you want to create the final build artifact, such as a JAR or WAR file, but don't need to install it into the local repository or use it as a dependency in other projects right away. |

1. Explain 3 types of maven repositories

Local repository: Maven local repository is located in your local system. It is created by the maven when you run any maven command.

Central repository: Maven central repository is located on the web. It has been created by the apache maven community itself.

Remote repository: Maven remote repository is located on the web. Most of libraries can be missing from the central repository such as JBoss library etc, so we need to define remote repository in pom.xml file.

1. Add your codes and answer sheet to a directory named “maven-basic-training” and push it to your training github repository.