**4. Maven**

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

**Build tools like Maven are used to automate the building and management of software projects. They help streamline the build process and make it easier to manage dependencies, compile source code, run tests, and package the application for deployment**

1. Install maven
2. Display output of maven version

**Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)**

**Maven home: C:\Program Files\apache-maven-3.8.5**

1. What is the pom.xml file?

**The** pom.xml **(Project Object Model) file is a central configuration file in Maven that defines the build and project configuration for a Java project. It contains information about the project's dependencies, build settings, and other metadata.**

**This file acts as a single point of reference for all build-related information, allowing developers to easily manage and maintain their projects.**

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

|  |  |
| --- | --- |
| groupId | Identifies the group or organization that is responsible for the project. It is used to group related projects together. |
| artifactId | Uniquely identifies the project within the group. |
| version | Specifies the version number of the project. |
| packaging | Specifies the type of artifact that the project produces, such as a JAR file or a WAR file. |
| dependencies | Defines the libraries and other dependencies that the project needs to build and run. Each dependency is defined as a dependency element. |
| dependency | Defines a single library or other dependency needed by the project. It includes information such as the groupId, artifactId, and version of the dependency, as well as any configuration options. |
| properties | Defines values for properties that can be used throughout the pom.xml file |

1. Create a method which accept an integer as parameter and returns the square of it
2. Add junit (v5) dependency

**JUnit is a unit testing framework for the Java programming language.**

**JUnit 5 is the next generation of JUnit. The goal is to create an up-to-date foundation for developer-side testing on the JVM. This includes focusing on Java 8 and above, as well as enabling many different styles of testing.**

1. Add a unit test to test the method
2. Run unit tests with maven. What is the command you used?

**mvn test**

1. 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

**Lombok is a Java library that can automatically generate boilerplate code. By using Lombok, can reduce the amount of code you have to write, making your code more concise and easier to maintain. Lombok can be added as a dependency to the project and then automatically generates the required code when compiling the code**

1. Explain the usage of these commands.

These Maven commands used to build and manage Java projects.

|  |  |
| --- | --- |
| mvn clean | Deletes the target directory and all generated files, effectively cleaning the project and preparing it for a fresh build. |
| mvn install | Compiles the source code, runs any tests, and installs the resulting artifact in the local Maven repository. This makes the artifact available for use as a dependency in other projects. |
| mvn package | Compiles the source code, runs any tests, and packages the resulting artifact into a distribution format such as a JAR or WAR file. This is useful for deploying the artifact to production environments or for distributing it to other developers. |

1. Explain 3 types of maven repositories

**Local repository**: A local repository is a cache of libraries and dependencies stored on the developer's local machine. When Maven builds a project, it first checks the local repository for the required dependencies. If they are not found, Maven downloads them from remote repositories and stores them in the local repository.

**Remote repository**: A remote repository is a centralized storage location for dependencies that are available to multiple developers or organizations. Remote repositories can be hosted by Maven itself or by third-party providers.

**Central repository**: The central repository is a remote repository that is hosted by the Maven community. It contains a large number of common Java libraries and other dependencies that can be used by Maven projects. When building a project, Maven first checks the local repository for the required dependencies, and if they are not found, Maven searches the central repository for the dependencies.

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

**A JAR file is a package file format typically used to aggregate many Java class files and associated metadata and resources into one file for distribution. JAR files are archive files that include a Java-specific manifest file**