**4. Maven**

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

Build tools like Maven is used primarily to streamline the process of building software in aspects of

* Dependency Management:

Maven provides robust dependency management capabilities, allowing you to declare project dependencies in the pom.xml file, automatically downloading and managing the correct versions of dependencies, and resolving conflicts between dependencies.

* Build Automation:

When using Maven the build process is stored in a configuration file (pom.xml) and use Maven commands to trigger the build process, making it easy to build and rebuild your software in a reproducible manner.

* Consistent Project Structure:

Maven project structure includes conventions for organizing source code, resources, tests, and build-related files, making it easier to locate and manage project artifacts.

1. Install maven
2. Display output of maven version

Command: mvn -version

Output:

Apache Maven 3.9.1 (2e178502fcdbffc201671fb2537d0cb4b4cc58f8)

Maven home: C:\Program Files\apache-maven-3.9.1

Java version: 17.0.6, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17

Default locale: en\_US, platform encoding: Cp1252

OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

1. What is the pom.xml file?

Pom.xml file is an XML file which is used to store configuration and metadata information for the project such as dependencies, build directory, source directory, test source directory or plugins.

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

|  |  |
| --- | --- |
| groupId | The group ID is a unique identifier that helps identify the project within a larger organization or project structure.  Eg: com.apache.maven |
| artifactId | The artifact ID is a unique identifier that helps identify the project's artifact, such as a JAR or WAR file, within the project's group.  Eg: commns-math |
| version | Version is used to specify the version number of the project's artifact. The version number can be in any format, but it is typically structured as a sequence of numbers separated by dots, such as "1.0.0", "2.3.1", or "0.9.8". |
| packaging | The packaging type determines the format in which the project's artifact will be built and packaged, such as a JAR (Java Archive), WAR (Web Archive), or POM (Project Object Model). |
| dependencies | Dependencies is used to define external libraries or modules that are required by the project's source code to compile, run, or perform other tasks. Maven automatically resolves and downloads them during the build process |
| dependency | Defines a single dependency of a Maven project. It specifies the details of an external library or module that the project depends on, such as its groupId, artifactId, version, and optional additional details. |
| properties | Properties in Maven are like variables that can be defined once and referenced in multiple places within the pom.xml file, allowing for centralization and reuse of common 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?

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
3. Explain the usage of these commands

|  |  |
| --- | --- |
| mvn clean | Used in Maven to clean the build artifacts and generated files from a previous build. It is often used as a preparatory step before starting a new build or when troubleshooting build-related issues. |
| mvn install | Used in Maven to build and package a project, and then install the resulting artifacts into the local Maven repository. |
| mvn package | Used in Maven to build and package a project, creating a distributable artifact such as a JAR or WAR file, depending on the packaging type specified in the project's pom.xml file. |

1. Explain 3 types of maven repositories

Maven repositories are used to store and manage artifacts (e.g., JAR, WAR, and other files).

* Local Repository:

The local repository is a local cache on your computer where Maven stores downloaded dependencies and locally-built artifacts.

* Remote Repository:

Remote repositories are repositories that are hosted on remote servers and accessed over the internet.

* Central Repository:

The Central Repository is the default and official repository provided by the Apache Maven project. It is a public remote repository that hosts a wide range of open-source Java artifacts, including libraries, frameworks, and plugins, that can be used as dependencies in Maven projects.

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