1. **What is Maven and why is it used?**

**Ans.**

Maven is a build automation and project management tool.

Maven is a build automation and project management tool used for Java projects. It simplifies dependency management, build processes, and project documentation.

Used for compiling, testing, packaging, and deploying applications.

Manages dependencies automatically via repositories.

Uses a POM.xml file to define project structure, dependencies, and configurations.

Maven has different features

1. Dependency management
2. Plugins
3. Standard structure layout
4. goals
5. **Explain the POM file in Maven.**

**Ans.**

POM stands for Project Object model

POM is an XML file

POM file maintains all the dependencies and configurations of the project

The POM.xml file includes the information about the project, build instructions

POM file should be unique

1. **What are Maven coordinates and what do they represent?**

Maven coordinates uniquely identify a project or dependency. They include:

* groupId: Unique identifier for an organization (e.g., org.springframework).
* artifactId: Name of the project or library (e.g., spring-core).
* version: The version of the artifact (e.g., 5.3.0).
* packaging (optional): Type of output (jar, war, ear).

1. **How do you manage dependencies in Maven?**

**Ans.**

Using the Repositories

Dependencies are defined in the pom.xml file within the <dependencies> section.

Maven downloads dependencies from repositories and stores them locally.

1. **What is a Maven repository and what are its types?**

**Ans.**

**Maven Repository:** Maven repository contains the dependencies

We have three types of Repositories

1. **Local repository:** Local repository available in our own systems/laptops

It is located in user/.m2

1. **Central repository:** central repository is belongs to the maven. It contains the all dependencies
2. **Remote repository**: If you are not find the dependency in local repository and central repository, it will available in the remote repository

1. **Explain the concept of Maven lifecycle phases.**

**Ans.**

**We have 3 phases in lifecycle**

**Default:**

In default phase we have few stages:

1. Validate: Validate the project structure
2. Compile: compile the source code
3. Test: test the source code
4. Run: run the source code
5. Verify: verify the project
6. Deploy: after verifying deploy the project

**Clean:**

Pre-clean: Removes the jar/war/ear files

Clean: cleans the jar/war/ear files

Post clean: generate the new jar/war/ear files

**Site:**

Pre-site: generate the pre site files from the post clean files

Site: generate the site files from the pre-site files

Post-site: generate the post site files from site file

Site-deploy: deploy the post site files

1. **What are Maven goals and how do they differ from phases?**

**Ans.**

**Maven goals:**

1. Clean: remove the target directory
2. Validate: check the project structure before build the project
3. Compile: compiles the source code
4. Test: run the unit tests using the testing frameworks
5. Package: all the files into single package
6. Install: installing the all the dependencies what project needs
7. Verify: Verify the project
8. Deploy: deploy the project
9. **How do you create a Maven project?**

**Ans.**

mvn archetype: generate

this is the command to create maven project

1. **What is a Maven plugin and how is it used?**

**Ans**.

Plugins have the project dependencies and libraries

mvn plugin\_name:goal

A **plugin** extends Maven’s functionality.

1. **How do you handle versioning in Maven projects?**

**Ans.**

Using the SNAPSHOTs

1. **Explain the PEM file in maven?**
2. **when we generate jar/war/ear file in target?**

**Ans.**

After running the mvn package command

1. **what is home directory in maven?**

**Ans.**

/.m2

1. **where will be build files stored?**

**Ans.**

Target directory

1. **what is meant by build tool?**

**Ans.**

Build tool provides the complete structure of the project and provide the dependencies and libraries, plugins. Developers no need to put efforts to create the project structure.

1. **Explain the process of building in maven?**

* Validate
* Compile
* Test
* Package
* Install
* Deploy

1. **Does maven support all types of projects to build?**

**Ans.** Yes

1. **what is the difference between compile & validate?**

**Compile:** compiles the source code of the project

**Validate:** checks the project structure before build the project

1. **can you create only one jar file or can we create multiple, explain?**

Yes, multiple JARs can be created using profiles or classifiers.

1. **What is Git and why is it used?**

Git stands for Global Information Tracker

Git is a version control system

Git maintains the versions history

Git is used for collaboration and track the changes

1. **Explain the difference between Git and other version control systems.**

**Git:**

Git keeps the history of the versions

Git supports the collaboration

Track the all changes

Git is a Decentralized control version system

**Git** is distributed, while SVN is centralized.

Git allows offline commits.

1. **How do you initialize a Git repository?**

Git init command is used to initialize the empty git repository

1. **What is the purpose of the .gitignore file?**

Excludes files from Git tracking.

1. **How do you stage changes in Git?**

Git add filename command is used to stage the all the changes in git

1. **What is the difference between git commit and git commit -m?**

git commit: Opens an editor for message(multiple lines).

git commit -m "Message": Adds a message inline.

1. **How do you create a new branch in Git?**

Git branch <branch\_nm> is the command to create a new branch

1. **What is the difference between git merge and git rebase?**

**Git merge:** git merge command is used to combine the changes from one branch to another branch

**Git rebase:** git rebase command is used to move the changes from one branch to another branch

**Merge**: Creates a new commit.

**Rebase**: Rewrites commit history.

1. **How do you resolve merge conflicts in Git?**

A merge conflict occurs when Git cannot automatically combine changes from two different branches. This usually happens when:

The same line in a file has been modified in both branches.

A file has been deleted in one branch but modified in another.

1. **What is the purpose of git stash?**

Git stash is used to temporarily save and store the files without committing

1. **Explain the use of git pull and git fetch.**

**Ans.** Git pull is used to merge and fetch the data from remote repository to local repository

**Git fetch:** git fetch command is used to fetching the data