Maven Quiz Paper

1)What is Maven ? What is your experience with it?

Maven is a project management and comprehension tool. Maven provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

My experience with maven is very well because by using maven i can usually do so many thing very easily which are very much task i had to do like for example my using maven i don't need to download all the jar file or dependencies here simply i will create pom.xml which will do everything for me automatically another thing my using maven i can easily compile build run and deploy my project.

2)What are the different tools that you have integrated Maven with?

I have integrated different kind of tools with Maven like Jenkins, chef etc

3)What are the steps that happen before a software projects deployment?

Following are the phases –

validate – validate the project is correct and all necessary information is available.

compile – compile the source code of the project.

test – test the compiled source code using a suitable unit testing framework.

These tests should not require the code be packaged or deployed

package – take the compiled code and package it in its distributable format, such as a JAR.

integration test – process and deploy the package if necessary into an environment where integration tests can be run.

verify – run any checks to verify the package is valid and meets quality criteria.

install – install the package into the local repository, for use as a dependency in other projects locally.

4)What are the Advantages of maven over Ant ?

Ant is simply a toolbox whereas Maven is about the application of patterns in order to achieve an infrastructure which displays the characteristics of visibility, reusability, maintainability, and comprehensibility. It is wrong to consider Maven as a build tool and just a replacement for Ant.

5)What does it mean when you say Maven uses Convention over Configuration?

Maven uses Convention over Configuration which means developers are not required to create build process themselves. Developers do not have to mention each and every configuration details

6)How do you know the version of mvn you are using?

mvn —version

7)List two most important files in your maven installation?

settings.xml file

pom.xml file

8)What is POM? What does it contain? Explain in detail

When you execute a Maven command you give Maven a POM file to execute the commands on. Maven will then execute the command on the resources described in the POM.
A Maven POM file (Project Object Model) is an XML file that describe the resources of the project. Th includes the directories where the source code, test source etc. is located in, what external dependencies (JAR files) your projects has etc.
The POM file describes what to build, but most often not how to build it. How to build it is up to the Maven build phases and goals. You can insert custom actions (goals) into the Maven build phase if y need to, though.
Each project has a POM file. The POM file is named pom.xml and should be located in the root direct of your project. A project divided into sub projects will typically have one POM file for the parent project and one POM file for each subproject. This structure allows both the total project to be built in one or any of the sub projects to be built separately.
Throughout the rest of this section I will describe the most important parts of the POM file. For a full reference of the POM file,

9)What is settings.xml ? What does it contain?

Maven has two settings files. In the settings files you can configure settings for Maven across all Mav POM files. For instance, you can configure:
Location of local repository Active build profile
Etc.
The settings files are called settings.xml. The two settings files are located at: The Maven installation directory: $M2_HOME/conf/settings.xml
The user's home directory: ${user.home}/.m2/settings.xml
Both files are optional. If both files are present, the values in the user home settings file overrides the
values in the Maven installation settings file.

10)What is a goal in Maven terminology?
A goal represents a specific task which contributes to the building and managing of a project. It may be bound to zero or more build phases. A goal not bound to any build phase could be executed outside of the build lifecycle by direct invocation.

11)Explain what the following commands will do?
mvn clean
is used to clean target folder
when mvn clean command executes, Maven deletes the build directory
mvn test
Run tests using a suitable unit testing framework (Junit is one).
mvn install
Install the package into the local repository, which can be used as a dependency in other projects locally.

mvn deploy
Copies the final package to the remote repository for sharing with other developers and projects.

12)What are the most important phases of a Maven Build Lifecycle?

A Build Lifecycle is a well-defined sequence of phases, which define the order in which the goals are to be executed. Here phase represents a stage in life cycle. As an example, a typical Maven Build Lifecycle consists of sequence of phases.

validate = Validating the information
Validates if the project is correct and if all necessary information is available.

package = packaging
This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml.

13)What are Maven Profiles ? How do you use a specific profile?

A Build profile is a set of configuration values which can be used to set or override default values of Maven build. Using a build profile, we can customize build for different environments such as Production v/s Development environments.
Profiles are specified in pom.xml file using its activeProfiles/profiles elements and are triggered in variety of ways. Profiles modify the POM at build time, and are used to give parameters different target environments (for example, the path of the database server in the development, testing, and production environments).

14) What are Maven Plugins? List a few important maven plugins that you used?
(Must talk about maven release plugin)
Maven is actually a plugin execution framework where every task is actually done by plugins.
Maven Plugins are generally used to:
create jar file
create war file
compile code files
unit testing of code
create project documentation
create project reports

The Maven does the following useful tasks using maven-release-plugin.
mvn release:clean
It cleans the workspace in case the last release process was not successful.
mvn release:rollback
Rollback the changes done to workspace code and configuration in case the last release process was not successful.
mvn release:prepare
mvn release:perform
mvn release:prepare
mvn release:perform

Build plugin : They execute during the build process and should be configured in the <build/> element of pom.xml.

15)How do you use plugins in Maven?
Maven is actually a plugin execution framework where every task is actually done by plugins. Maven Plugins are generally used to:
create jar file
create war file
compile code files
unit testing of code
create project documentation
create project reports

16)What are dependencies in maven? How are they declared? How is transitive dependencies managed in Maven?
One of the first goals Maven executes is to check the dependencies needed by your project. Dependencies are external JAR files (Java libraries) that our project uses. If the dependencies are n found in the local Maven repository,
Directly specify the versions of artifacts to be used when they are encountered in transitive dependencies. For an example project C can include B as a dependency in its dependency Management section and directly control which version of B is to be used when it is ever referenced.

17)What is difference between Dependency Management and Plugin Management in maven?

One of the core features of Maven is Dependency Management. Managing dependencies is a difficult task once we've to deal with multi-module projects (consisting of hundreds of modules/ sub-projects). Maven provides a high degree of control to manage such scenarios.
Usually, we have a set of project under a common project. In such case, we can create a common pom having all the common dependencies and then make this pom, the parent of sub-project's poms.
If we take a look at the various plugin configurations, we can see the HSQLDB dependencies duplicated in several places. Unfortunately, dependencyManagement doesn't apply to plugin dependencies, but we can still use a property to consolidate the versions. Most complex Maven multimodule projects tend to define all versions in the top-level POM. This top-level POM then becomes a focal point for changes that affect the entire project.

19)How would you if a given pom is parent pom or a child pom ?
Maven use the effective POM (configuration from super pom plus project configuration) to execute relevant goal. It helps developers to specify minimum configuration detail in his/her pom.xml. Although configurations can be overridden easily.

An easy way to look at the default configurations of the super POM is by running the following command: mvn help:effective-pom.

20)How would you use modules in maven? When do you create a module for your project?

A large software application generally consists of multiple modules and it is common scenario where multiple teams are working on different modules of same application. For example, consider a team is working on the front end of the application as app-ui project (app-ui.jar:1.0) and they are using data-service project (data-service.jar:1.0).

21)What is the order of inheritence between following files:

settings file
project pom
child/ module pom
company pom