|  |  |
| --- | --- |
|  |  |
|  | **Maven**  1- What is MAVEN and its usage. |
|  |  |
|  | ans- Maven was originally started as an attempt to simplify the build processes in the Jakarta Turbine project. |
|  | Maven is able to build any number of projects into predefined output types such as a JAR, WAR, or |
|  |  |
|  | distribution based on metadata about the project, without any need to do any scripting in most |
|  | cases. |
|  |  |
|  | Maven's primary goal is to allow a developer to comprehend the complete state of a development effort |
|  | attempts to deal with: |
|  | • Making the build process easy in the shortest period of time. |
|  | In order to attain this goal there are several areas of concern that Maven |
|  | • Providing a uniform build system |
|  | • Providing quality project information |
|  | • Providing guidelines for best practices development |
|  | • Allowing transparent migration to new features |
|  |  |
|  | 2- How to create maven project? |
|  |  |
|  | ans- The Eclipse window opens on the screen. Since there aren’t any projects yet, complete the following steps: |
|  |  |
|  | Go to the File option |
|  | In the drop-down menu, select New |
|  | Select the Project option. |
|  | Select the Maven Project option |
|  | Click on Next |
|  | A dialog box will appear. Select the default workspace. |
|  |  |
|  | Click on “Next” |
|  |  |
|  | Select a plugin there and click on “Next” |
|  | In the next dialog box that appears |
|  |  |
|  | Enter the Group ID |
|  | “12345” |
|  |  |
|  | Enter the Artifact ID |
|  | “mavenproject” |
|  |  |
|  | Click on “Finish” |
|  | The project is now created. |
|  |  |
|  | 3- Maven Lifecycle' |
|  |  |
|  | ans- The default Maven lifecycle consists of 8 major steps or phases for compiling, testing, building and installing a given Java project as specified below: |
|  |  |
|  | Validate: This step validates if the project structure is correct. For example – It checks if all the dependencies have been downloaded and are available in the local repository. |
|  | Compile: It compiles the source code, converts the .java files to .class and stores the classes in target/classes folder. |
|  | Test: It runs unit tests for the project. |
|  | Package: This step packages the compiled code in distributable format like JAR or WAR. |
|  | Integration test: It runs the integration tests for the project. |
|  | Verify: This step runs checks to verify that the project is valid and meets the quality standards. |
|  | Install: This step installs the packaged code to the local Maven repository. |
|  | Deploy: It copies the packaged code to the remote repository for sharing it with other developers. |
|  |  |
|  | 4- maven cammands. |
|  |  |
|  | ans- mvn clean: Cleans the project and removes all files generated by the previous build. |
|  | mvn compile: Compiles source code of the project. |
|  | mvn test-compile: Compiles the test source code. |
|  | mvn test: Runs tests for the project. |
|  | mvn package: Creates JAR or WAR file for the project to convert it into a distributable format. |
|  | mvn install: Deploys the packaged JAR/ WAR file to the local repository. |
|  | mvn deploy: Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project. |