Q1) What is Maven?

Answer: Maven is a project management and understanding tool. Maven provides developers with a complete life-style design. With the use of a standard directory structure and the default lifespan cycle, the development team can build the structure of the project at any time.

Q2) How does Maven manage?

Answer: Maven provides ways for developers to manage the following:

- Builds
- Documentation
- Reporting
- Dependents
- SCMs
- Publications
- Distribution
- Mail list

Q3) How do you know the version of MVN you use?

þÿ A n s w e r : E n t e r t h e f o l l o w i n g c o m m a n d

þÿ m v n v e r s i o n

Q4) What is balm?

Answer: The project object refers to the POM model. This is the base unit that works in Maven. This is an XML file. It always lives as a pom.xml in the base directory of the program. The project (s) contains information about the project used and the structure of various structures used by Maven.

Q5) What is the Maven occupation?

Answer: An artistic file is a file, usually sent to a JAR repository by JAR. Creates a masterpiece þÿ o f o n e o r m o r e a r t i f a c t s , s u c h a s c o m p i l e d J A R a n d s o u r c e s J A R . E a c h ID (usually the reverse domain name, com.example.foo), a satellite ID (a name) and a version string. Three find a unique identity together. Functions of a project are represented as

complications.

Q6) What is a storehouse?

Answer: A directory that contains the compiled JAR file pom.xml file as a soft repository. Maan is þÿlooking for dependents in the barns. & local repository. Central reposito

Q7) What is the Maven area repository?

þÿAnswer: Local repository. Maven s local repository is a folder of the dev the projects are stored locally. When Maven is implemented, Maven automatically downloads all the independent jars in the local store. This folder is usually named. M2.

Q8) What is the structure profile?

Answer: A configuration profile is a set of configuration values that can be used to set or ignore the default values of the Maven Building. Using the creation details, you can customize it to various environments, such as production v / s development environments.

Q9) What are the different architectural profiles?

þÿAnswer: There are three types of create profiles

þÿA project project POM file, defined by pom.xml.

þÿA user defined mammon in the xml file (% USER_HOME% / .m2 / settin

þÿGlobal Manual Universal systems are defined in the xml file (% M2_HO settings.xml)

Q10) What is a Maven Store?

Answer: There is a repository that can be stored in jazz, library jar, plug-in or any other specific artwork in all the projects, and Maven can easily be used.

Q11) What is the local repository?

Answer: Maven is a folder on your computer as a local repository. This will be created when the þÿfirst time you execute any superior command. Maven s local repository k functions (library jars, plugin jars etc.).

Q12) What is the default location for your local repository?

Answer: ~ / M2 / repository.

Q13) What is the command to install the JAR file in the local repository?

Answer: mvn install

Q14) What is Central Storage?

Answer: This is a repository provided by the Maven community. These are commonly used libraries. If Maven does not see any kind of local repository in the local repository, the following URL begins to search in the central repository: http://repo1.maven.org/maven2/.

Q15) What is Arctic?

Answer: The arctic is a dynamic plugin and its mission is to create a project structure according to its template.

Q16) What is the command to create a new project based on a hard drive?

Answer: Enter the following command -mvn archetype: create

Q17) What does dependency management mean in the dependable function environment?

Answer: They represent versions of the straightforward components, which must be used when interacting with intermediate functions. For example, project B may be dependent on behalf of B. Direct control of the admin control unit and any version of B controlled directly.

Q18) What is the default value for the packaging element?

Answer: Valid packaging values are some jar, war, ear and go. If the packaging value is not specified, it will automatically run to the jar.

Q19) What is the full name of a project?

Answer: <GroupId>: <artifactId>: <version>

Q20) What is a prohibition?

þÿAnswer: Any sort of relativity can be removed using the delete elemen are excluded C because B and P are dependent on C.

Q21) What is the central repository of Maven?

Answer: Maven is a repository of words, which can easily be used in all projects, library jar, plug-ins or any other definite art galleries. The Maven Storage is of three types.

Q22) What is the use of Settings XML in Maven?

Answer: A maven settings.xml file defines the values that configure the Maven function in different ways. Usually, it is used to define a local storage space, alternate remote storage servers, and authentication information for individual repositories.

þÿQ23) What s Maven Central Gradually?

Answer: A module dependency generates the output of a particular function on a module behalf. Volumes are usually stored in a repository, such as Maven Central, a corporate maven or repository or a directory in a local file system.

Q24) Is Maven available to Java?

Answer: Ungrounded objects can be challenged and removed. Maven is primarily an automation tool used for Java projects. Maven will store Java libraries and Maven plugins in one or more repositories as a central repository and store them on a local cache.

Q25) How to override the default name of the war file ?

Answer: You can use the element <finalName></finalName> in your pom.xml file to override the default name of the war file.

Q26) What is element used to specify the dependencies in your pom.xml ?

Answer:

<dependency>

</dependency>

Q27) If we build the parent project in a multi module setup Maven will build all the child projects/modules ?

Answer: TRUE

Q28) Using which element in the child modules pom.xml do we refer to the parent pom ?

Answer:

<parent>

</parent>

Q29) Using which element does maven allows us specify the visibility

of dependencies ?

Answer:

<scope>

</scope>

Q30) What are six types of scopes ?

Answer:

- Compile
- Runtime
- Provided
- Test
- System
- Import

Q31) What compile type scope ?

Answer: If we use compile type scope, then those dependencies will be available during the project build, that is when the classes are compiled, tests are compiled, when the tests, applications run.

Q32) What is default scope that the maven uses if we do not specify scope element in pom.xml ?

Answer: Compile

Q33) What happens when we declare dependencies under the scope provided ?

Answer: It means that those are the dependencies which are required during build test and run but they are not required to be exported, meaning they need not be part of the application when it is deployed.

Q34) What scope can be used to make certain dependencies to be available only during compiling and running tests ?

Answer: Test

Q35) Example of dependency which can be declared under test scope ?

Answer: junit.

Q36) What is maven profiles?

Answer: Maven profiles are used to setup environment specific configuration files. We need to use the profile tag in pom.xml file to setup environment conf files.

Q37) Difference between Ant and Maven.

Answer:

- Ant and Maven are both build tools from Apache company.
- Maven is a framework and ant is tool
- Maven having lifecycles, ant wont having.
- Maven scripts are stored pom.xml file and ant scripts are stored in build.xml
- Maven plugins are reusable and ant tags are reusable.

Q38) Explain about the Maven repositories.

Answer: There are three types of maven repositories.

Local repository

Central repository

Remote repository

While downloading the jars, First it will look into local , central and remote . it will download the jars and use to compile code in the above order.

Q39) Explain about the maven plugins.

Answer: Maven is the plugin based framework. Each activity in maven depend on plugin. All the

þÿmaven plugins are stored in the group org.apache.maven.plugins .

Ex: To compile java code, we need to use maven-jar-plugin

þÿTo create jar file, we need to use maven-jar-plugin so on.

Q40)What is maven dependencies?

Answer: Maven dependencies are used to compile the code.

All the dependencies are called in the between the tags in pom file

<dependencies></dependencies>

Q41) Explain about the maven lifecycles.

Answer:

clean is used to delete the existing target directory.

Validate is used to check the syntax of the maven pom file.

Compile is used to compile the source code

Install is used to create a jar /war/ear file.

Q42) Explain about the system and provided.?

Answer: <scope>system</scope> is used to represent the local jars inside the pom file.

<scope>provided</scope> is used to avoid the jar from the WEB-INF/lib folder of the war.

Q43) What is archetype.

Answer: We will use maven archetype to generate the maven folder structure based on the inputs.

Ex: mvn archetype:generate.

Q44) What is the maven pom?

Ans: Maven pom stands for project object model.Every pom file must contains, projectId , artifcatId, version and packaging tags. And also we need to call all the dependencies in between the dependencies tag.

<dependencies></dependencies>

Along with we need to call <build></build> tag to build the code.

Q45) What is the version to use to check maven version

þÿ A n s w e r : m v n v e r s i o n

Q46) Explain about the parent pom file.

Answer: We can use the maven parent pom file to call the other modules.

Q47) What maven modules?

Answer: we can use maven modules in the parent pom file as a module.

<modules>

</modules>

Each module having separate pom file.

Q48) Which is the command will use to add jar manually in pom file under dependency.

Answer: <systemPath></systemPath>

Q49) Expain about the maven settings.xml file?

Answer:

It contains the below configuration

- Prody configuration
- Local repository configuration
- Remote repository configuration
- Central Repository configuration

Q50) In which way the dependencies jar files are downloaded from net.

Answer:

- First it will the jar in local, if available it will take from local and compile the code.
- If the jar is not available in local repo, then it will got central repo and download the jar and will compile
- It the jar is not available in central , it will go remote repo and download the jar and compile the code .

If the jar is not available in three places, maven will failed the build.

Q51)What is the advantage of Maven?

Answer:

Easy to download the Jar file using Dependency

Configuration setup is easier

Project View

Q52) what are the types of POM configuration?

Answer: Mail List, Plugins, goals, projects version and build profile.

Q53) what is GroupId refereces?

Answer: GroupId is nothing but reverse domain access and it will referes which file location

Q54) what is ArtifactsId refereces?

Answer: ArtifactsId is nothing but file name to download in accessed domain and it will referes file name.

Q55) what is Version and Scope refereces?

Answer: Version which defines the version control and scope which referes the file name.

Q56)What does the mvn clean command?

Answer: mvn clean command is used to removes the target location file of before and after build starts.

Q57) what is Maven Repository referred?

Answer: where the place we can see all the project jars,plugins and library jar file and Store easily in maven

Q58)what are the types of Maven Plugins?

Answer:

- Report Maven Plugin
- Build Maven Plugin

Q59)Define MOJO?

Answer: A MOJO defines as Maven plain Old Java Object. it is an executable goal of maven plan.

Q60) Maven differentiation with Selenium webdriver?

Answer:

- In selenium web driver, invalid jar files are added to the project as library files, whereas in maven the jar files are added as dependencies.
- In Selenium Web driver, for every individual project, it is essential to add all the jar files again to the new project, whereas this can be overcome by using maven by adding a pom.xml file to the new project.

Q61)How to exclude dependencies?

Answer: By using the exclusion element. Exclusions are specific dependency in your POM, and are targeted at a specific groupId and artifactId.

Q62) which command removes the target directory before the start of

a build process?

Answer: mvn clean command

Q63 Processes involved in achieving the project?

Answer: Maven helps in creating a Java-based project more easily. Accessibility of new feature created or added in Maven added to a project in Maven configuration.

Q64)which Languages are supported by maven?

C#,

Ruby,

Scala,

other languages

Q65)pom.xml file?

Answer: Collections of dependencies instead of jar files with group ID and artifact ID

Q66) How pom helps maven?

Answer: POM is a page object model. It represents the basic unit of work in maven. POM contains the configuration as

build profiles.

project version.

developers.

mailing list.

project dependencies.

plugins.

goals

Q67) what is clean, default and site in maven?

Answer:

Clean represents cleans up artifacts achieved by previous builds.

Default (or build): this can be customized to build the appliance.

Site: generates site level documents for the project.

Q68) how does test and package phases implies in maven?

Answer:

Test þÿ" t h e c o m p i l e d s o u r c e c o d e i n v o l v e s t h e a p p r o p r i a t e u n i t t e s t i n g f
not required the code deployed or packaged.

Package þÿ" t a k e t h e c o m p i l e d c o d e a n d p a c k a g e i n a p a r t i c u l a r f o r m a t l i k

Q69) Features of build tool?

Answer:

Compile Java code and build jar for deployment and release.

Versioning and dependencies management

Run to look at and report test results

Run code analyse check with Sonar, check style, Find bugs, etc.

Q70) runtime in maven ?

Answer: It represents that the dependency will not be required for compilation

Q71)How does Maven manage?

Answer: Maven provides ways for developers to manage the following:

- Builds
- Documentation
- Reporting
- Dependents
- SCMs
- Publications
- Distribution
- Mail list

Q72)How do you know the version of MVN you use?

þÿ A n s w e r : E n t e r t h e f o l l o w i n g c o m m a n d m v n v e r s i o n

Q73) What is balm?

Answer: The project object refers to the POM model. This is the base unit that works in Maven. This is an XML file. It always lives as a pom.xml in the base directory of the program. The project (s) contains information about the project used and the structure of various structures used by Maven.

Q74)What is the Mayan occupation?

Answer: An artistic file is a file, usually sent to a JAR repository by JAR. Creates a masterpiece of one or more artifacts, such as compiled JAR and "sources" JAR.Each artificial

group is a group ID (usually the reverse domain name,com.example.foo), a satellite ID (a name) and a version string. Three find a unique identity together. Functions of a project are represented as complications.

Q75)What is a storehouse?

Answer: A directory that contains the compiled JAR file pom.xml file as a soft repository.

Maven is looking for dependents in the

- barns
- local repository
- Central repository
- Remote repository

Q76) What do you know about Maven?

Answer: Maven is actually a tool for Project management and Comprehension from Apache, which supports the users to create a framework for lifecycle. Using the inner piece of information, Maven could be able to manage new project creation, reports and documentation. Maven can be used to manage and create any project based on Java. The project build infrastructure in Maven can be automated by a team of developers quite simply and quickly because Maven uses a common build lifecycle and a standard directory. Maven is capable of setting up the working method standards too quickly while developers working in different development team environment. Maven is very much user friendly as most of the project setups are quite easy and reusable.

Q77) What are the primary objectives of Maven?

Answer: The primary objective of maven is to support the developer with key features following:

- To provide the developer an extensive model to set up projects to make it reusable, easily

maintainable and easily understandable.

To support with tools or plug-ins which interact with the demonstrative model.

Q78) What is the command to know which version of Maven you are using?

Answer: The command to know the version of Maven is:

þÿ $ m v n v e r s i o n

Q79) What are the support Maven offers to the developers?

Answer: Maven supports the developers to control and administer:

Project creations (builds)

Reports

Documentation

SCMs

Dependencies

Mailing lists

Releases and

Distribution

Q80) What do you mean by POM in maven?

Answer: POM in Maven refers to Project Object Model, which is the principal object model for a work or a project. POM has an XML file which contains the information about the project and other configuration information to create a project in Maven.

Q81) What are the configurations that can be specified in POM?

Answer: The following are some of the configurations that can be specified in POM:

- Plug-ins
- Build Profiles
- Project Dependencies
- Mailing Lists
- Project Version
- Goals

There are some prerequisites before creating POM. The developer has to decide the project group i.e group ID, name of the project (artifact ID) and the version. These attributes help the developers to identify the project in your repository.

Q82) What is the use of Maven?

Answer: Maven was actually built to make the build processes present in Jakarta Turbine projects easier and simpler. Further, Maven was enhanced by Apache to create various projects together, publish the information of projects and to share the JAR files present in various projects so as to support in the association of teams.

Q83) What is the difference between Convention and Configuration and state why Maven uses Convention?

Answer: In case of Configuration, developers need to create the build processes manually and has to specify each and every configuration in detail. But, Maven uses convention instead of configuration, where the developers need not create the build processes, whereas Maven does it automatically. Also, for convention, users need not specify the configuration in details. Once a developer creates a project in Maven, it will automatically create a structure. Developers have to place the files appropriately. There is no need to specify any configuration details in pom.xml.

Q84) State the features present in Maven?

Answer: The following are some of the features present in Maven:

- þÿ M a v e n f o l l o w s v e r y s i m p l e p r o j e c t s e t u p t h e r e b y s i m p l i f y i n g d e v e l o p e
- Steady utilization of resources among all the projects.
- Frequently updated vast collection of libraries
- Automatically updates it.
- Dependency management
- User can extend with plug-ins in Java or some other programming languages
- It permits support and access immediately to latest features with little or no configuration.

Q85) How will you install Maven on to your machine?

Answer: Installing Maven requires installation of Java because Maven is a tool based on Java.

þÿ M a v e n c a n b e i n s t a l l e d i n W i n d o w s , L i n u x a n d M a c a s p e r t h e u s e r s w i s

basic requirements to install Maven in your machine:

- Install JDK 1.7 or higher on to your machine. Verify Java installation on your machine.
- Set Environment variables for Java
- Download Maven Archive from Apache website. There are separate archives available for windows, Linux and Mac OS. Download the archive as per your need.
- Extract the Maven archive.
- Set Environment variables for Maven.
- Add the Maven bin directory location to the system path.
- þÿ V e r i f y t h e M a v e n i n s t a l l a t i o n u s i n g t h e c o m m a n d : $ m v n v e r s i o n

Q86) Explain the use of Maven artifact?

Answer: Maven artifact is a Jar file present in the Maven repository. With one or multiple artifacts in a Maven build produces a compiled Jar or sources of Jar. Every artifact will contain a group ID, version number and an artifact ID.

Q87) State the differences between Apache Ant and Maven?

Answer:

Apache Ant	Maven
Apache ant is basically a tool box	Maven is a framework
Apache Ant will not support any formal conventions such as project directory structure	Maven supports formal conventions
Apache Ant does everything procedurally	Maven does things in a declarative way
It is not possible to reuse the scripts of Ant	The plug-ins of Maven were reusable
Ant will not contain lifecycle. User has to manually add the sequence of tasks about to be performed	Maven contain a life cycle

Q88) How many POMs can be created for a project? Explain the minimum requirements for POM?

Answer: Only one POM file can be created for a project. For every POM file, project element, group id, artifact id and version are mandatory. Every project has to be stored in the repository like the following notation: groupid:artifactid:version. There are some basic requirements for a POM. They are:

- Root of the project: Root tag helps you to check the primary specifications such as apache schema and w3.org.
- Version of the model: The model version should be 4.0.0 or higher.
- Group Id: Group id is the id for a project group. Usually these IDs were named unique. For example, the group id for a insurance company can be com.company.insurance which contains all projects related to insurance.

Artifact Id: Artifact Id is the id for a particular project, usually the name of the project. Artifact Id and Group Id specifies the location of artifact in the repository.

Version of the project: To avoid confusions among different versions, version of the project has to be specified. Say for example,

com.company.insurance:consumer:insurance:1.1

com.company.insurance:consumer:insurance:1.2 and so on.

Q89) What do you mean by Super POM?

Answer: The default POM of Maven is the Super POM. All POMs can be derived from a parent or by default. This POM is the basic POM and is also known as super POM. Super POM will contains default derived values. To execute the desired objective, Maven use the effective POM. Super POM supports the developers to configure pom.xml with a minimum configuration. These configurations can be easily modified. The following command is used to check the default configurations of super POM:

$mvn help:effective-pom

Q90) How will you create an effective POM on your computer?

Answer: Initially, create a pom.xml in some directory of your computer. Specify minimum configuration on your computer. Check the default configuration of the effective POM through the following command:

$mvn help:effective-pom

Say for an example, create pom.xml in C:MVN/Project Folder

Then, open the command prompt (windows), navigate to the folder of pom.xml, execute mvn command.

C:MVN/Project >mvn help:effective-pom

Now, Maven will begin to process and the effective POM can be visualized in the console after the successful application of interpolation, inheritance and profiles.

Q91) What do you mean by build life cycle? Explain the phases of build life cycle?

Answer: A build life cycle can be defined as a decisive order of phases. Phases define the sequence in which the desired goals to be accomplished. Phase refers to a stage in the lifecycle. A distinctive Maven build life cycle will comprise of the following order of phases.

Prepare-Resources: In this phase, user can customize resource copying.

Compile: This phase compiles the source code

Validate: This phase validates the project if the project and other information provided were correct.

Test: This phase tests the compiled source code which is suited for testing the framework.

Package: Creates the package JAR/WAR .

Install: Installs the package into remote or local repository of Maven.

Deploy: This phase copies the final package into the remote repository.

Q92) What do you mean by Goal in Maven life cycle?

Answer: Pre-phase and Post-phase goals will be present and these goals run before or after a specific phase. Once Maven begins to build a project, it follows the already defined order of phases and executed. Goal refers to a particular task which assists to build and manage a project. There could be no goals or it could build more phases. The sequence of execution depends on the order of goals and the build phases are called.

Q93) List the standard life cycles of Maven?

Answer: There are three standard lifecycles available for Maven. They are:

Clean

Build (default)

Site

Q94) Explain Clean life cycle in Maven?

Answer: While executing the mvn post-clean command, Maven calls the clean life cycle. It consists of the following phases:

Clean

Pre-clean

Post-clean

The goal of clean phase in Maven lies within the clean life cycle. Clean:clean goal thus removes the build output by deleting the build directory. Once mvn clean command is executed, maven deletes the build directory. Developer can also customize the role of goals by specifying goals in any phase of the clean life cycle.

Q95) Explain Default life cycle in Maven?

Answer: As the name implies, default is the basic life cycle of Maven to build the application. Default lifecycle contains 21 phases namely validate, initialize, generate-sources, generate-resources, process-sources, process-resources, compile, process-classes, process-test sources, generate test-sources, process-test-classes, test, test-compile, package, prepare-package, pre-integration-test, post-integration-test, integration- test, verify, install and deploy.

Q96) Explain Site life cycle in Maven?

Answer: Site lifecycle in Maven is a plugin which is normally used to make new documentation to create reports, to deploy a site and many more. The site life cycle consists of the following phases:

Site

Pre-site

Post-site

Site-Deploy

Q97) What will happen once a Maven command is passed to run a phase?

Answer: Once a phase is invoked through Maven command, (say for example mvn test), till that phase and inclusive of the called phase only will get executed.

Q98) What do you mean by build profile in Maven?

Answer: Build profile in Maven refers to a set of configuration values through which user can set a default value or can override the default values of Maven build. User can customize build for several environments using a build profile. Profiles could be able to modify the POM at build time. They are also used to provide parameters for various target environments.

Q99) What are the types of Build profile in Maven?

Answer: The build profile of Maven is of three types. They are:

Per-project: It is specified in the project file i.e POM file, pom.xml

Per-user: Specified in the Maven settings XML file(settings.xml-can be accessed from root folder)

Global: Specified in the Maven global settings XML file

Q100) What are the ways to activate a Maven build profile?

Answer: The build profile of Maven can be activated using several ways. They are:

Through settings of Maven

- Explicitly using the command prompt (command console)
- Based on user or system variables (environment variables)
- Operating System settings
- Through current files or missing files.

Q101) What do you mean by Maven repository?

Answer: Maven repository is the directory which stores jar files such as project jar file, library jar file, plug-ins or some project specific artifacts. Maven accesses this directory and uses it quite easily.

Q102) What are the types of Maven Repository?

Answer: The following are the types of Maven repository. They are:

- Local repository
- Central repository
- Remote repository

Q103) Explain Maven Local Repository?

Answer: Local repository is the local folder located in your machine. Once you run any of the Maven command for the first time, local repository gets created on to your machine. All the project related dependencies such as library.jar, plugin.jar etc., were stored in this repository. Once you run a maven build, maven starts downloading all the dependency jar files stored in the local repository. Every time you build a project, local repository avoids references to the dependencies. User can also override the default location of the local repository.

Q104) Explain Maven Central Repository?

Answer: Central repository in Maven is a repository from the Maven community which consists of several number of libraries which are generally used. If Maven is unable to locate any

dependency in the local repository, it begins to search in the central repository through the following URL:

www.repo1.maven.org/maven2/

Central repository does not need any configuration. To search the repositories, it needs internet access.

Q105) Explain Maven Remote Repository?

Answer: If Maven does not locate a dependency in local repository or central repository, it further searches the remote repository. Remote repository is a customized repository provided to the developer with the necessary library files and other project relevant jar files.

Q106) Explain the Dependency Search Sequence in Maven?

Answer: Once the developer execute s the Maven build commands, Maven begins to look for dependency libraries in the following steps:

Search for a dependency in the local repository. If the dependency is not found, move to step 2 or else perform further processing.

Search for a dependency in the central repository. If the dependency is not found and if remote repository/ repositories explicitly mentioned by the developer, move to step 4. Else download the dependency to the local repository for future use.

þÿ I f t h e d e v e l o p e r d o e s n t m e n t i o n a r e m o t e r e p o s i t o r y , M a v e n s t o p s t o

simply throws an error, which means the searched dependency is not found.

If the dependency searched by an user is found in the remote repository/ repositories, that will be downloaded to the local repository and can be used for future references. Else, Maven throws an error, which means the searched dependency is not found.d

Q107) What do you mean by Maven Plug-ins?

Answer: Literally, Maven is a plug-in execution framework. It executes each and every task through plug-ins. They are commonly used to :create jar/war files, to compile source code files, do unit testing of the source code, create project reports and documentation. Usually plug-ins provide some set of goals. The same can be executed through the following command:

$mvn [Name of the plug-in]:[Name of the Goal]

Q108) What are the types of Maven Plug-ins?

Answer: There are two types of Plug-ins from Maven. They are:

Build plug-ins: This type of plug-in is used to execute while build process is in progress and has to be configured in the <build/> element of pom.xml file.

Reporting plug-ins: This type of plug-ins were executed while the site generation process is in progress and has to be configured in <reporting/> element of the pom.xml file.

Q109) What are the common Plug-ins available in Maven?

Answer: Maven supports some other common Plug-ins namely: Clean, Compiler, JAR, WAR, Surefire, Javadoc and antrun.

Q110) How will you create a project in Maven?

Answer: Creating a project in Maven requires installation of Java JDK on to your machine. To create projects, Maven uses archetype plug-ins. Similarly, to create a basic java application, we shall use maven-archetype-quickstart plug-in. As an example, we shall create a Maven based Java application project in the local machine in some folder, C:\MVN. Navigate to the command prompt, go to C:\MVN directory and execute the mvn command that follows:

C:\MVN>mvn archetype:generate

DgroupId = com.companyname.insurance

DartifactId = consumerInsurance

DarchetypeArtifactId = maven-archetype-quickstart

DinteractiveMode = False

Maven will now start to process and creates the complete java application project structure.

Q111) What is the need for external dependencies in Maven?

Answer: The dependency management in Maven is managed by repositories. In case, if we search for a dependency and if it is not available in any of the repositories such as local, central and remote, Maven allows a user to use External dependency. Through external dependency, user can add the dependency they wish. For an example, if we are creating a java application þÿ a n d w e d o n t h a v e t h e d e p e n d e n c y i n a n y o f t h e r e p o s i t o r i e s a n d c r e a t i n external dependency, add lib folder to the src folder. Copy any jar to the lib folder. The jar you have copied is the jar created by the user itself. So, external dependency is used to add your own jar files to the library.

Q112) How will you create an external dependency?

Answer: The following are the steps to create external dependency:

Configure pom.xml in the similar fashion as we used to configure other dependencies.

Mention the groupId name similar to the name of the library

Mention the artifactId name similar to the name of the library

Define the scope of the system

Define the system path as per the location of the project.

Add lib folder to the src folder

Copy your own jar file to the lib folder.

Q113) How will you create project documents in Maven? Explain with an example.

Answer: As an example, we shall create a simple java application. The application can be stored in the following directory: C:/MVN. Create your application and store in this directory.

C:\MVN\mvn default . Maven will begin to build the project stored in the directory. Once after compilation and execution, Maven will create all necessary report documentation.

Q114) What do you mean by archetype in Maven?

Answer: Archetype is yet another plug-in available in Maven which creates a project structure as per the template specified to it. Arche type can be accessed using the following command:

$mvn archetype:generate

Q115) What are the different archetypes available in Maven?

Answer: Maven supports various archetypes. They are:

- maven-archetype-archetype
- maven-archetype-mojo
- maven-archetype-j2ee-simple
- maven-archetype-plugin-site
- maven-archetype-plugin
- maven-archetype-simple
- maven-archetype-quickstart
- maven-archetype-portlet
- maven-archetype-site

maven-archetype-site-simple

þÿ m a v e n - a r c h e t y p e w e b a p p

Q116) What do you mean by Maven Snapshots?

Answer: Snap shot is a special version available in the remote repository of Maven. It indicates the latest development copy. For every build, Maven checks for a new version of Snapshot in the remote repository. The data service team is responsible to update snapshot with updated source code each and every time to the repository for every build of Maven.

Q117) What is the difference between version and Snapshot?

Answer: Version and snapshots are different from one another. If Maven downloaded some version namely info-service:1.1, Maven never try to download a newer file in the name of 1.1 to the repository. If it needs to update, the new version will be saved as 1.2. But, in case of a snapshot, Maven will fetch the latest available snapshot for every build of the project automatically.

Q118) What do you mean by build automation in Maven?

Answer: Build automation is used to specify the scenario while dependent project or projects were begun after the successful completion of the project build to affirm the dependent project(s) is/are stable.

Q119) What do you mean by dependency mediation?

Answer: When several versions of an artifact were applied, dependency mediation determines which version of a dependency has to be used. If in case we have two dependency versions at the same depth in the dependency tree, the dependency which is initially declared will be used.

Q120) What do you mean by dependency management?

Answer: Dependency management specifies which version of artifact to be used while they are encountered in transitive dependencies. Say for example, project D could be able to add C as a

dependency in the dependency management section and controls directly what version of C as to be used when it is referenced ever.

Q121) What is the scope of transitive dependencies discovery?

Answer: The scope of transitive dependencies discovery is restricted by several scope of dependencies. They are:

Compile: Compile refers that the dependency is present on the classpath of the project. Compile is the default scope.

Provided: This scope specifies that the dependency has to be supplied by JDK or Container/Web server during runtime.

Runtime: Specifies that dependency is not needed to compile but needed during execution.

Test: This scope specifies that the dependency is available for test compilation and execution alone.

System: Specifies that the user has provided the system path

Import: Used only if the dependency is of the type POM. This scope specifies the POM has to be replaced with the dependencies in that POM using <dependencymanagement> section.

Q122) What are the steps involved in the deployment process?

Answer: The deployment process in a project development generally will comprise of the following steps:

Check the source codes from all the projects in progress to the version control system, SVN or the source code repository and tag it.

Then, download the entire source codes from the SVN

Now, start building the application

Save the build output in EAR or WAR file in a common network location

Collect the file from the network and then deploy that file to production

Now, update the documentation using the date and the upgraded version number of the project application.

Q123) What is the use of SCM in pom.xml file?

Answer: SCM is used to configure the location of SVN from the place where Maven will check for the source code.

Q124) How will you run test classes in Maven? Specify.

Answer: Developer require a plug-in called surefire to run test classes in Maven. Set up that plugin, verify and configure the settings present in the pom.xml and setting.xml and check for a þÿ p r o p e r t y c a l l e d t e s t .

Q125) What are the configurations possible in Settings.xml file?

Answer: The following configurations are possible in Settings.xml file:

User can configure the proxy

User can configure Local repository

User can configure central repository

User can configure remote repository