ASSIGNMENT

1. Write Maven life cycle and its commands

Maven:

Maven is a tool that makes easier for developers when developing reports, checks build and testing automation setups and used for projects build, dependency and documentation based on POM (Projecct Object Model).

Life Cycle:

Maven generally has 8 steps in it's default lifecycle.

1.Validate - Validates if the project structure is correct or not
2.Compile - Compiles the source code
3.Test - Runs unit tests
4.Package - Packages the compiled code in the formats like war/jar/pom
5.Integration Test - Runs integration tests for the project
6.Verify - Verifies the project is valid or not and meets the standards
7.Install - Installs the packaged code to the local Maven Repository
8.Deploy - Copies the packaged code to remote repository for sharing

2. Describe 3 lines each

mvn --version : Prints out the version of Maven you are running
mvn -- compile: Compiles source code of the project
mvn --test : Complies the test source code
mvn --install : Builds the project described by your Maven POM file and installs the resulting artifact (JAR) into your local Maven repository
mvn --clean : Clears the target directory into which Maven normally builds your project.

3. What is Web Services?

Web service is a technology to communicate one programming language with another. A web service is a software system that supports interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically, web Service Definition Language, or WSDL). web services fulfill a specific task or a set of tasks.

4. What is Rest Controller?

RestController is a Spring annotation that is used to build REST API in a declarative way. RestController annotation is applied to a class to mark it as a request handler, and Spring will do the building and provide the RESTful web service at runtime.

5. Describe Web Services and its methods

A web service is a software system that supports interoperable machine-to-machine interaction over a network.

1.GET - Provides a read only access to a resource from databases

2.POST - Creates a new resource or insert the record to databases

3.DELETE - Used to remove a resource from databases.

4.PUT - Used to update a existing resource or create a new resource.

6. GIT Life Cycle

1.You clone the Git repository as a working copy.

2.You modify the working copy by adding/editing files.

3.If necessary, you also update the working copy by taking other developer's changes.

4.You review the changes before commit.

5.You commit changes. If everything is fine, then you push the changes to the repository.

6.After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.

7. Describe the GIT Commands

git init - This command is used to create a local repository.

git add - This command is used to add one or more files to staging (Index) area.

git commit - Commit command is used in two scenarios.

1. **Git commit -m :** This command changes the head. It records or snapshots the file permanently in the version history with a message.
2. **Git commit -a :** This command commits any files added in the repository with git add and also commits any files you've changed since then

.

git pull - Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

git push - It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo.

git checkout - The **git checkout** command is used to switch between branches in a repository.

git checkout -b - The git checkout -b option is a convenience flag that performs run git branch <new-branch>operation before running git checkout <new-branch>.

git checkout -d - Rather than checking out a branch to work on it, check out a commit for inspection and discardable experiments. This

is the default behavior of `git checkout <commit>` when `<commit>` is not a branch name.

git log - This command is used to check the commit history.

git reset - This command unstages the file, but it preserves the file contents.

git revert - The git revert command is used for undoing changes to a repository’s commit history.

git merge - This command is used to merge the specified branch?s history into the current branch.

git rebase - Rebasing is a process to reapply commits on top of another base trip.

8. Centralized v/s Distributed version control

1.Centralized version control(CVS):

• In CVS, a client need to get local copy of source from server, do the changes and commit those changes to centeral source on server.

• CVS systems are easy to learn and set up.

• Working on branches in difficult in CVS. Developer often faces merge conflicts.

2.Distributed version control(DVS):

• In DVS, each client can have a local branch as well and have a complete history on it. Client need to push the changes to branch which will then be pushed to server repository.

• DVS systems are difficult for beginners. Multiple commands needs to be remembered.

• Working on branches in easier in DVS. Developer faces lesser conflicts.