Git Interview

What is Git and why is it used?

Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to work on a project simultaneously, keeping track of their changes and merging them seamlessly.

What is the difference between Git and GitHub?

Git is a version control system, whereas GitHub is a platform that hosts Git repositories and provides collaboration features such as issue tracking, pull requests, and project management tools.

Explain the difference between Git merge and Git rebase.

Git merge integrates changes from one branch into another, creating a merge commit to record the merge. Git rebase, on the other hand, rewrites the commit history by moving the commits of one branch onto another, resulting in a linear history without merge commits.

What is a Git repository?

A Git repository is a storage location where a Git project's metadata and object database are stored. It contains all the files and directories along with their complete version history.

What is a Git branch and why would you create one?

A Git branch is a separate line of development that diverges from the main codebase (usually from the master branch). Branches are used to work on new features, bug fixes, or experiments without affecting the main codebase until changes are ready to be merged.

How do you resolve merge conflicts in Git?

Merge conflicts occur when Git is unable to automatically merge changes from different branches. To resolve conflicts, you need to manually edit the conflicting files to resolve differences, then add the resolved files and commit the changes.

What is Git stash and how is it used?

Git stash is used to temporarily store changes that are not ready to be committed but need to be removed from the working directory. It allows you to switch branches or perform other operations without committing the changes. Stashed changes can later be reapplied or permanently discarded.

Explain the Git workflow you follow in your projects.

This would vary depending on the individual or team's preferred workflow. Common workflows include Centralized Workflow, Feature Branch Workflow, Gitflow Workflow, and Forking Workflow. The answer should include details on how branches are used, how changes are merged, and any specific conventions or tools utilized.

What are some Git best practices you follow?

Some best practices include committing small, focused changes, writing descriptive commit messages, keeping branches and commit history clean, regularly pulling changes from remote repositories, and using Git hooks for automating tasks like code linting or testing.

How do you revert a commit in Git?

Git provides several ways to revert a commit, depending on the situation. One common method is to use git revert <commit-id> to create a new commit that undoes the changes introduced by a specific commit. Another approach is to use git reset to reset the HEAD to a previous commit, either keeping the changes in the working directory (--mixed), staging area (--soft), or discarding them entirely (--hard).

How do you initialize a Git repository in a Java project?

You can initialize a Git repository in a Java project by navigating to the project directory in the terminal and running the command git init.

Explain how you would add a new Java class file to a Git repository.

To add a new Java class file to a Git repository, you would first create the file in your project directory. Then, you would stage the file for commit using the command git add <filename> and commit the changes using git commit -m "Add new Java class file".

What are some common Git ignore patterns for Java projects?

Common Git ignore patterns for Java projects include ignoring build artifacts (e.g., target/ directory), IDE-specific files (e.g., .idea/, .classpath, .project for IntelliJ IDEA or Eclipse), and dependencies (e.g., \*.jar, \*.war).

How do you check the status of changes in a Git repository for a Java project?

You can check the status of changes in a Git repository by running the command git status. This command will show you which files have been modified, staged, or are untracked.

Explain the process of branching and merging in Git within the context of a Java project.

Branching in Git allows you to create a separate line of development for new features or bug fixes. In a Java project, you might create a new branch using git branch <branch-name> and switch to it using git checkout <branch-name>. After making changes in the branch, you would merge it back into the main branch (e.g., master) using git merge <branch-name>.

How do you handle dependencies in a Java project managed with Git?

In a Java project managed with Git, dependencies are typically managed using build automation tools like Maven or Gradle. The dependencies are declared in the project's pom.xml (Maven) or build.gradle (Gradle) file, which are version-controlled along with the rest of the project.

What is a Git submodule and how might you use it in a Java project?

A Git submodule is a reference to another Git repository within a parent Git repository. Submodules are useful in Java projects when you need to include external libraries or dependencies that are managed in separate Git repositories.

Explain the difference between Git pull and Git fetch in the context of a Java project.

git pull fetches changes from a remote repository and merges them into the current branch, while git fetch only downloads changes from the remote repository but does not merge them. In a Java project, you might use git pull to update your local repository with changes from the remote repository.

Git Commands :

git init: Initializes a new Git repository in the current directory.

git clone [url]: Clones a remote repository into a new directory.

git add [file]: Adds a file or changes in a file to the staging area.

git commit -m "Message": Commits staged changes to the local repository with a descriptive message.

git status: Displays the status of the working directory and staged changes.

git diff: Shows the differences between the working directory and the staging area.

git diff --staged: Shows the differences between the staged changes and the last commit.

git log: Displays the commit history.

git branch: Lists all branches in the repository.

git branch [branch\_name]: Creates a new branch with the specified name.

git checkout [branch\_name]: Switches to the specified branch.

git checkout -b [branch\_name]: Creates a new branch and switches to it.

git merge [branch\_name]: Merges changes from the specified branch into the current branch.

git pull: Fetches changes from the remote repository and merges them into the current branch.

git push: Pushes local commits to the remote repository.

git remote -v: Lists all remote repositories and their URLs.

git remote add [name] [url]: Adds a new remote repository with the specified name and URL.

git remote rm [name]: Removes the remote repository with the specified name.

git rm [file]: Removes a file from the working directory and stages the change for commit.

git mv [old\_name] [new\_name]: Renames a file and stages the change for commit.

git stash: Stashes changes in the working directory for later use.

git stash list: Lists all stashed changes.

git stash apply: Applies the most recent stash to the working directory.

git stash pop: Applies the most recent stash and removes it from the stash list.

git reset [commit]: Resets the HEAD to the specified commit.

git checkout -- [file]: Discards changes in the working directory for the specified file.

What is a version control system (VCS)?

A VCS keeps track of the contributions of the developers working as a team on the projects. They maintain the history of code changes done and with project evolution, it gives an upper hand to the developers to introduce new code, fixes bugs, and run tests with confidence that their previously working copy could be restored at any moment in case things go wrong.

What does git add command do?

This command adds files and changes to the index of the existing directory.

You can add all changes at once using git add . command.

You can add files one by one specifically using git add <file\_name> command.

You can add contents of a particular folder by using git add /<folder\_name>/ command.

How will you create a git repository?

Have git installed in your system.

Then in order to create a git repository, create a folder for the project and then run git init.

Doing this will create a .git file in the project folder which indicates that the repository has been created.

Tell me something about git stash?

Git stash can be used in cases where we need to switch in between branches and at the same time not wanting to lose edits in the current branch. Running the git stash command basically pushes the current working directory state and index to the stack for future use and thereby providing a clean working directory for other tasks

What is the command used to delete a branch?

To delete a branch we can simply use the command git branch –d [head].

To delete a branch locally, we can simply run the command: git branch -d <local\_branch\_name>

To delete a branch remotely, run the command: git push origin --delete <remote\_branch\_name>

Deleting a branching scenario occurs for multiple reasons. One such reason is to get rid of the feature branches once it has been merged into the development branch

What is a conflict?

Git usually handles feature merges automatically but sometimes while working in a team environment, there might be cases of conflicts such as:

1. When two separate branches have changes to the same line in a file

2. A file is deleted in one branch but has been modified in the other.

These conflicts have to be solved manually after discussion with the team as git will not be able to predict what and whose changes have to be given precedence.

What command helps us know the list of branches merged to master?

git branch --merged helps to get the list of the branches that have been merged into the current branch.

Note: git branch --no-merged lists the branches that have not been merged to the current branch