1. 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 collaborate on projects simultaneously and efficiently manage changes to the codebase.

2. Explain the difference between Git pull and Git fetch.

Git pull: This command fetches changes from a remote repository and integrates them into the current branch. It's essentially a combination of git fetch and git merge.

Git fetch: This command only fetches changes from the remote repository to the local repository, updating the remote tracking branches. It doesn't modify the working directory or the current branch.

3. How do you revert a commit in Git?

git revert <commit-hash>
This command creates a new commit that undoes the changes made by the specified commit.

4. Describe the Git staging area.

The staging area, also known as the index, is where you prepare changes before committing them to the Git repository. It acts as a middle ground between your working directory and the repository.

5. What is a merge conflict, and how can it be resolved?

A merge conflict occurs when Git is unable to automatically resolve differences in code between two branches being merged. This often happens when changes are made to the same part of a file on different branches.

To resolve a merge conflict, you need to manually edit the conflicting files to resolve the differences, then add the resolved files to the staging area, and finally commit the changes.

6. How does Git branching contribute to collaboration?

Each developer can create a separate branch for their changes, work independently, and later merge their changes back into the main branch or any other relevant branches.

7. What is the purpose of Git rebase?

Git rebase is used to integrate changes from one branch into another by rewriting the commit history. It helps in maintaining a cleaner and linear project history compared to traditional merge commits.

8. Explain the difference between Git clone and Git fork.

Git clone: This command creates a local copy of a remote repository, allowing us to work on the project locally and push changes back to the original repository if we have permission.

Git fork: Forking creates a copy of a repository on a remote server (like GitHub), under your own account. It's typically used when you want to contribute to a project for which you don't have write access.

9. How do you delete a branch in Git?

git branch -d <branch-name>

To delete a branch forcefully (regardless of its merge status)
git branch -D <branch-name>

10. What is a Git hook, and how can it be used?

Git hooks are scripts that Git executes before or after events such as commits, merges, and pushes. They allow you to automate tasks or enforce policies within a Git repository. Git hooks are stored in the .git/hooks directory of your repository.