

### **1. What is Git and why is it used?**

Git is a version control system that tracks changes in files, allowing multiple people to work on the same project without overwriting each other's work. It's used for source code management in software development to keep track of code history, collaborate with others, and revert to previous versions if needed.

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

**Git pull** updates the local repository with changes from the remote repository and merges them into the current branch. **Git fetch** retrieves changes from the remote repository but doesn't merge them, allowing you to review the changes before integrating them.

### **3. How do you revert a commit in Git?**

To revert a commit, use **git revert <commit-hash>**. This creates a new commit that undoes the changes made by the specific commit without altering the existing history.

### **4. Describe the Git staging area.**

The staging area, or index, is where you prepare changes before committing them. You add changes to the staging area using **git add**, and then commit them with **git commit**, making the changes part of the repository history.

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

A merge conflict occurs when Git can't automatically merge changes from different branches because the same part of a file was edited differently. Resolve it by manually editing the conflicting files to keep the desired changes and then committing the resolved changes.

## **6. How does Git branching contribute to collaboration?**

Git branching allows multiple people to work on different features or bug fixes simultaneously without affecting the main codebase. Each branch is an isolated environment, making it easier to manage and integrate different changes when they're ready.

## **7. What is the purpose of Git rebase?**

Git rebase integrates changes from one branch into another by moving or combining a sequence of commits. It creates a cleaner project history by making it appear as if the changes were made in a linear sequence.

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

**Git clone** creates a local copy of a remote repository, while **Git fork** creates a personal copy of someone else's repository on GitHub. You can then clone your forked repository to work on it locally.

## **9. How do you delete a branch in Git?**

To delete a branch, use `git branch -d <branch-name>` for a local branch. For a remote branch, use `git push origin --delete <branch-name>`.

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

A Git hook is a script that runs automatically at certain points in the Git workflow, such as before a commit or after a push. They can be used to enforce coding standards, run tests, or perform other automated tasks to streamline the development process.

Charan Sai  
Meta Scifor