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

Git is a version control system which lets us track changes we make in the files over time. With Git, we can revert to various states of your files. We can also make a copy of your file, make changes to that copy, and then merge these changes to the original copy. It provides a way for multiple developers to collaborate on a project, manage and merge changes efficiently, and maintain a history of all modifications made to the codebase. Git saves all the history of the files and also the changes made in it.

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

**git pull**

Git pull command is used to fetch and merge any commits from the tracking remote branch.

**git fetch [alias]**

Git fetch command is used to fetch down all the branches from that Git remote.

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

Initially use the command **git log** to get the details of all the commits made and check for which commit should be reverted and then use **git revert** to undo the changes made in that commit.

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

A file in the staged state means it is ready to be committed. Once all the necessary changes have been made, so the next step is to move the file to the commit state.

**git status**

show modified files in working directory, staged for your next commit

**git add [file]**

add a file as it looks now to your next commit (stage)

**git reset [file]**

unstage a file while retaining the changes in working directory

**git diff**

diff of what is changed but not staged

**git diff --staged**

diff of what is staged but not yet commited

**git commit -m “[message]”**

commit your staged content as a new commit snapshot

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

Merge conflicts happen when users make different changes to the same line of the same file, or when one person edits a file and another person deletes the same file. Then we must resolve all merge conflicts before we can merge a pull request on GitHub.

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

By doing git branching, a user can collaborate with others or multiple users who are within the same team. So that all the individuals can work on a single project from different places remotely by branching. By creating a branch and doing work will keep things safer and can be done efficiently.

* Fetch and merge changes from the remote.
* Create a branch to work on a new project feature.
* Develop the feature on a branch and execute the work.
* Fetch and merge from the remote again in case new commits were made
* Push branch up to the remote

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

**git rebase [branch]**

Apply any commits of current branch ahead of specified one

The purpose of **git rebase** in Git is to modify the commit history by moving, combining, or deleting existing commits.

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

A fork is a copy of a repository that allows you to make your own changes without impacting the original project. Git clone creates a linked copy that will continue to synchronize with the target repository.

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

To delete, use the command git branch -d <branch\_name>

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