1. What is Git?
    - Answer: Git is a distributed version control system (DVCS) used to track changes in source code during software development. It allows multiple developers to work on a project simultaneously without interfering with each other's changes [1].
2. What is a repository in Git?
    - Answer: A Git repository (or repo) is a file structure that stores all the files for a project. It tracks changes made to these files over time, facilitating collaboration among team members [1].
3. What is the difference between Git and GitHub?
    - Answer: Git is a version control system used to track changes in files over time, while GitHub is a platform where Git repositories can be stored and shared. Git operates locally on your computer, whereas GitHub is a cloud-based service
4. What is the purpose of the .gitignore file?
    - Answer: The .gitignore file tells Git which files and folders to ignore when tracking changes. This helps keep the repository clean by excluding unnecessary files like logs, temporary files, or compiled code.
5. What does the git clone command do?
    - Answer: The git clone command creates a copy (or clone) of an existing Git repository. It is typically used to get a copy of a remote repository to the local machine.
6. What does the git config command do?
    - Answer: The git config command is used to set configuration options for defining the behavior of the repository, user information, and preferences. For example, setting up your name and email address before using Git commands [2].
7. What is a branch in Git, and why is it used?
    - Answer: A branch in Git is a separate line of development. It allows you to work on different features or bug fixes independently of the main codebase. Branches help in

managing and merging different lines of development efficiently [3].

8. How do you resolve a merge conflict in Git?
    - Answer: To resolve a merge conflict in Git, you need to open the conflicting files and manually edit the parts of the code that are in conflict. After resolving the conflicts, you add the resolved files to the staging area and commit the changes
9. What is the difference between git pull and git fetch?
    - Answer: git pull fetches changes from a remote repository and immediately merges them into your current branch. git fetch, on the other hand, only downloads the changes from the remote repository but does not merge them. This allows you to review the changes before integrating them [1].
10. What is a commit in Git?
    - Answer: A commit in Git is a snapshot of your repository at a specific point in time. Each commit has a unique ID and contains information about the changes made, the author, and the date and time of the commit [2].
11. What is a detached HEAD state in Git?
    - Answer: A detached HEAD state occurs when Git's HEAD pointer is not pointing to the latest commit of the current branch. This can happen when you checkout a specific commit or tag instead of a branch [2].
12. What is rebasing in Git, and how is it different from merging?
    - Answer: Rebasing in Git is the process of moving or combining a sequence of commits to a new base commit. Unlike merging, which creates a new commit that combines the changes from two branches, rebasing rewrites the commit history to create a linear sequence of commits [3].
13. How do you create a new branch in Git?
    - Answer: You can create a new branch in Git using the git branch <branch-name> command. To switch to the new branch, use git checkout <branch-name> [3].
14. What is the purpose of the git stash command?

- **Answer:** The git stash command temporarily saves your changes in a stash, allowing you to switch branches or work on something else without committing the changes. You can later apply the stashed changes using git stash apply [3].

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

- **Answer:** Todeleteabranch in Git, usethe git branch -d <branch-name>command for local branches and git push origin --delete <branch-name> for remote branches [3].

16. **What is the difference between git reset and git revert?**

- **Answer:** git reset moves the HEAD pointer to a previous commit and canmodifythe commit history, potentially losing changes. git revert creates a new commit that undoes the changes of a previous commit, preserving the commit history [3].

17. **What is the difference between git merge and git rebase?**

- **Answer:** git mergecombines thechanges fromone branch into another, creating a new commit that represents the merge. git rebase moves or combines a sequence of commits to a new base commit, creating a linear history [1].

18. **How do you revert a commit that has already been pushed and made public?**

- **Answer:** To revert a commit that hasbeen pushed, you can usethe git revert <commit> command. This creates a new commit that undoes the changes made by the specified commit [2].

19. **What is the purpose of the git cherry-pick command?**

- **Answer:** The gitcherry-pickcommandallows youto applythe changes introduced by an existing commitontoyour current branch. This is useful for selectively applying specific commits from one branch to another [2].

20. **What is the difference between git stash pop and git stash apply?**

- **Answer:** Bothcommandsapply stashed changes to your working directory. However, git stash pop applies the changes and then removes the stash from the stash list, while git stash

apply applies the changes but keeps the stash in the stash list [3].

21. How do you handle large binary files in Git?

- Answer: Git is not optimized for handling large binary files. To manage large files, you can use Git Large File Storage (LFS), which replaces large files with text pointers inside Git while storing the actual file content on a remote server [3].

22. What is the purpose of the git bisect command?

- Answer: The git bisect command is used to find the commit that introduced a bug by performing a binary search through the commit history. This helps in identifying the exact commit where the issue was introduced [3].

23. How do you squash commits in Git?

- Answer: To squash commits, you can use an interactive rebase with the git rebase -i <base-commit> command. This allows you to combine multiple commits into a single commit, making the commit history cleaner [3].

24. What is the difference between git diff and git status?

- Answer: git diff shows the changes between commits, commit and working tree, etc., while git status displays the state of the working directory and the staging area, showing which changes have been staged, which haven't, and which files aren't being tracked by Git [3].

25. What is the purpose of the git tag command?

- Answer: The git tag command is used to create a tag in Git, which is a reference to a specific point in the repository's history. Tags are often used to mark release points (e.g., v1.0, v2.0) and are immutable [1].

26. How do you list all branches in a Git repository?

- Answer: To list all branches in a Git repository, you can use the git branch command for local branches and git branch -r for remote branches. To list both local and remote branches, use git branch -a [2].

27. What is the difference between git log and git reflog?

- Answer: git log shows the commit history of the current branch, while git reflog shows a history of all the changes

made to the tip of branches and other references in the repository, including actions like checkouts and resets [2].

28. How do you handle a situation where you need to make a change in a previous commit?

- Answer: To make changes to a previous commit, you can use the git commit --amend command if the commit is the most recent one. For older commits, you can use an interactive rebase (git rebase -i <commit>) to edit the commit [3].

29. What is the purpose of the git submodule command?

- Answer: The git submodule command is used to manage external repositories within a Git repository. Submodules allow you to keep a Git repository as a subdirectory of another Git repository, which is useful for including external dependencies [3].

30. How do you undo the last commit without losing the changes?

- Answer: To undo the last commit without losing the changes, you can use the git reset --soft HEAD~1 command. This moves the HEAD pointer to the previous commit and keeps the changes in the staging area[3].

31. What is the difference between git archive and git bundle?

- Answer: git archive creates a tar or zip file containing the contents of a repository at a specific point in time, while git bundle creates a single file that contains the entire repository or a subset of commits, which can be used to transfer the repository [3].

32. How do you configure a Git repository to ignore file mode changes?

- Answer: To configure a Git repository to ignore file mode changes, you can set the core.fileMode configuration to false using the command git config core.fileMode false [3].