1. **What is git’s feature?**

Git features version control, branching and merging, distributed development, collaboration tools like pull requests, and cryptographic integrity checks.

1. **What are the advantages of using git?**

Git provides version control, enables collaboration, supports branching and merging, ensures code integrity, and is highly flexible and widely supported.

1. **What are git’s limitation?**

Learning Curve

Large File Handling

Merge Conflicts

Lack of GUI for Some Operations

Risk of Data Loss

1. **What are branches, and what command creates a new branch?**

Branches in Git are parallel lines of development that diverge from the main line (usually called the "master" or "main" branch) to work on different features, fixes, or experiments independently. They allow developers to isolate their work from the main codebase until it's ready to be merged.

The command git branch <branch\_name> creates a new branch in Git.

1. **What is git merge?**

git merge is a command used to integrate changes from one branch into another branch. When you merge one branch (called the source branch) into another branch (called the destination branch), Git combines the changes made in the source branch with the changes in the destination branch.

1. **When should you use git push and git pull?**

You should use git push when you want to send your local commits to a remote repository, typically after making changes and committing them locally. This action updates the remote repository with your changes.

On the other hand, you should use git pull when you want to retrieve changes from a remote repository and integrate them into your local repository. This is typically done to update your local repository with changes made by others in the remote repository.

1. **Is Github an Open-Source Software?**

**GitHub itself is not open-source software, but it hosts millions of open-source projects. The platform provides proprietary services for managing Git repositories, issue tracking, project management, and collaboration tools. However, many of the projects hosted on GitHub are open source, meaning their source code is freely available for anyone to view, modify, and distribute according to the terms of an open-source license.**

1. **What Is Repo?**

"Repo" is short for "repository." In the context of Git and GitHub, a repository is a storage space where your project's files and revision history are stored. It contains all the files for your project, along with a history of each file's changes. Repositories can be either local (stored on your computer) or remote (stored on a server, like GitHub).

1. **What is GitHub fork?**

Forking on GitHub refers to creating a personal copy of someone else's repository. When you fork a repository, GitHub creates a copy of the original repository in your GitHub account. You can then freely experiment with changes in your fork without affecting the original project. Forking is commonly used to propose changes to someone else's project (via pull requests), to experiment with changes, or to create your own version of an existing project.

1. **Can we change our GitHub username?**

Yes, you can change your GitHub username.

1. **Is it possible in git to merge a branch into the master?**

yes, it is possible in Git to merge a branch into the master branch.

1. **What language is used in git?**

Git itself is primarily written in the C programming language. However, Git also includes scripts and tools written in various languages, including shell scripting (Bash), Perl, Tcl, and Python. These scripts and tools contribute to Git's functionality and support its operation across different platforms and environments.

1. **What is the significance of Git version control?**

Git version control allows tracking changes, enabling collaboration, managing code history, facilitating branching and merging, ensuring code integrity, and streamlining deployment.

1. **List the Types of version control system.**

Centralized Version Control Systems (CVCS): In CVCS, there is a central server that stores the repository, and developers check out and commit changes directly to this central repository. Examples include CVS (Concurrent Versions System) and SVN (Subversion).

Distributed Version Control Systems (DVCS): In DVCS, each developer has their own local copy of the repository, including the full history of the project. Developers can work independently and synchronize their changes with remote repositories as needed. Git is the most popular example of a DVCS.

1. **Can we delete the github account.**

Yes, you can delete a GitHub account.

1. **Name any two git repository hosting services that are common.**

Two common Git repository hosting services are Bitbucket and GitLab.

1. **What is git clone?**

git clone is a Git command used to create a copy of a repository from a remote source, such as GitHub, Bitbucket, or GitLab, to your local machine. It copies all the files, commit history, and branches from the remote repository to your local machine, allowing you to work on the project locally and synchronize changes with the remote repository.

1. **What is staging environment?**

A staging environment is a pre-production environment that closely resembles the production environment. It serves as an intermediate step between development and production environments, allowing developers to test their code in a controlled environment before deploying it to production.

1. **What is github issues?**

GitHub Issues is a feature provided by GitHub for tracking tasks, enhancements, bugs, or any other topics related to a repository. It allows users to create, assign, and track issues within a project. Each issue can have a title, description, labels, assignees, milestones, and comments.

1. **What is a git submodules and git aliases?**

Git submodules and Git aliases are two different features of Git:

Git Submodules: Git submodules allow you to include a separate Git repository as a subdirectory within your own repository. This is useful when you want to include another project as a dependency in your own project. Git submodules allow you to track the specific version of the dependent repository that your project relies on, making it easier to manage dependencies and collaborate with others.

Git Aliases: Git aliases are custom shortcuts or abbreviations for Git commands. They allow you to create shorter, more memorable names for frequently used Git commands or command sequences. For example, you could create an alias co for git checkout, ci for git commit, or st for git status. Git aliases can be defined either globally (for all repositories on your system) or locally (for a specific repository) in your Git configuration file. They help improve productivity and streamline your workflow by reducing the need to type out long Git commands repeatedly.