Basic Questions:

1. What is the difference between Git and SVN?

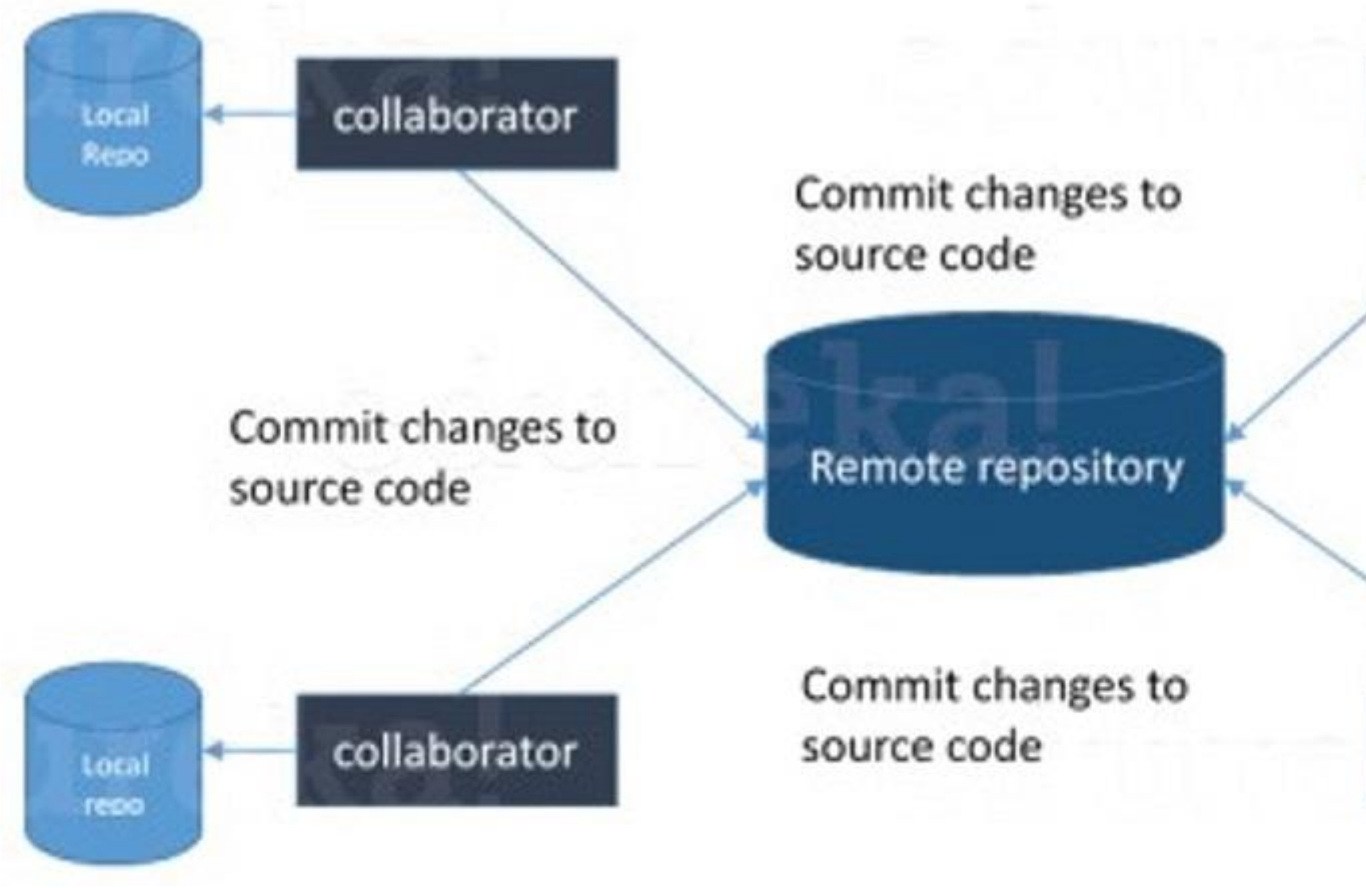
Git	SVN
Git is a Decentralized Version Control tool	SVN is a Centralized Version Control tool
It belongs to the 3rd generation of Version Control tools	It belongs to the 2nd generation of Version Control tools
Clients can clone entire repositories on their local systems	Version history is stored on a server-side repository
Commits are possible even if offline	Only online commits are allowed
Push/pull operations are faster	Push/pull operations are slower
Works are shared automatically by commit	Nothing is shared automatically

2. What is Git?

I will suggest you attempt this question by first telling about the architecture of git as shown in the below diagram just try to explain the diagram by saying:

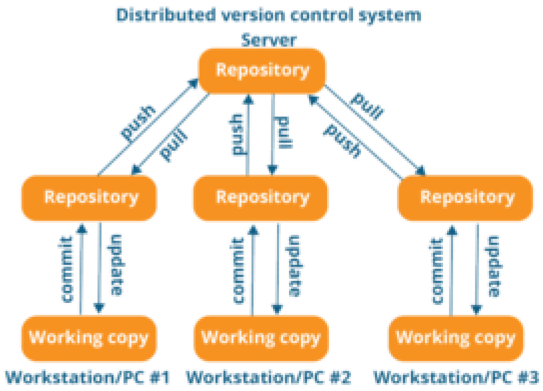
- Git is a Distributed Version Control system(DVCS). It lets you track changes made to a file and allows you to revert back to any particular change that you wish.
- It is a distributed architecture that provides many advantages over other Version Control Systems (VCS) like SVN. One of the major advantages is that it does not rely on a central server to store all the versions of a project's files.
- Instead, every developer "clones" a copy of a repository I have shown in the diagram with "Local repository" and has the full history of the project available on his hard drive. So when there is a server outage all you need to do to recover is one of your teammate's local Git repository.

- There is a central cloud repository where developers can commit changes and share them with other teammates.


Local Repo
collaborator
Commit changes to source code
Remote repository
Commit changes to source code
Commit changes to source code
Local repo
collaborator

3. What is a distributed VCS?

- These are the systems that don’t rely on a central server to store a project file and all its versions.
- In Distributed VCS, every contributor can get a local copy or “clone” of the main repository.
- As you can see in the above diagram, every programmer can maintain a local repository which is actually the copy or clone of the central repository which is present on their hard drive. They can commit and update their local repository without any hassles.
- With an operation called “pull”, they can update their local repositories with new data from the central server and “pull” operation affects changes to the main repository from their local repository.



4. What is the difference between Git and Github?

Git is a version control system of distributed nature that is used to track changes in source code during software development. It aids in coordinating work among programmers, but it can be used to track changes in any set of files. The main objectives of Git are speed, data integrity, and support for distributed, non-linear workflows.

GitHub is a Git repository hosting service, plus it adds many of its own features. GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, basic task management tools for every project.

5. What are the benefits of using Version Control System?

- With the Version Control System(VCS), all the team members are allowed to work freely on any file at any time. VCS gives you the flexibility to merge all the changes into a common version.
- All the previous versions and variants are neatly packed up inside the VCS. You can request any version at any time as per your requirement and you'll have a snapshot of the complete project right at hand.
- Whenever you save a new version of your project, your VCS requires you to provide a short description of the changes that you have made. Additionally, you can see what changes are made in the file's content. This helps you to know what changes have been made in the project and by whom.
- A distributed VCS like Git allows all the team members to have a complete history of the project so if there is a breakdown in the central server you can use any of your teammate's local Git repository.

6. What language is used in Git?

Instead of just telling the name of the language, you need to tell the reason for using it as well. I will suggest you to answer this by saying:

Git uses 'C' language. GIT is fast, and 'C' language makes this possible by reducing the overhead of run times associated with high-level languages.

7. Mention the various Git repository hosting functions.

- Github
- Gitlab
- Bitbucket
- SourceForge
- GitEnterprise

8. What is a commit message?

The command that is used to write a commit message is “`git commit -a`”. Now explain about -a flag by saying -a on the command line instructs git to commit the new content of all tracked files that have been modified. Also, mention you can use “`git add <file>`” before git commit -a if new files need to be committed for the first time.

9. How can you fix a broken commit?

In order to fix any broken commit, use the command “`git commit --amend`”. When you run this command, you can fix the broken commit message in the editor.

10. What is a repository in Git?

Repository in Git is a place where Git stores all the files. Git can store the files either on the local repository or on the remote repository.

11. How can you create a repository in Git?

This is probably the most frequently asked question and the answer to this is really simple.

To create a repository, create a directory for the project if it does not exist, then run the command “`git init`”. By running this command .git directory will be created in the project directory.

12. What is ‘bare repository’ in Git?

A “bare” repository in Git contains information about the version control and no working files (no tree) and it doesn’t contain the special .git sub-directory. Instead, it contains all the contents of the .git sub-directory directly in the main directory itself, whereas the working directory consists of :

1. A .git subdirectory with all the Git related revision history of your repository.
2. A working tree, or checked out copies of your project files.

13. What is a ‘conflict’ in git?

Git can handle on its own most merges by using its automatic merging features. There arises a conflict when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other. Conflicts are most likely to happen when working in a team environment.

14. How is git instaweb used?

‘`git instaweb’` is used to automatically direct a web browser and run a webserver with an interface into your local repository.

15. What is git is-tree?

‘`git is-tree`’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

16. Name a few Git commands and explain their usage.

Below are some basic Git commands:

Command	Function
`git rm [file]`	deletes the file from your working directory and stages the deletion.
`git log`	list the version history for the current branch.
`git show [commit]`	shows the metadata and content changes of the specified commit.
`git tag [commitID]`	used to give tags to the specified commit.

`git checkout [branch name]`
`git checkout -b [branch name]`
	used to switch from one branch to another.
	creates a new branch and also switches to it.

There can be two approaches to tackle this question and make sure that you include both because any of the below options can be used depending on the situation:

- Remove or fix the bad file in a new commit and then push it to the remote repository. This is the most obvious way to fix an error. Once you have made necessary changes to the file, then commit it to the remote repository using the command: git commit -m “commit message”
- Also, you can create a new commit that undoes all changes that were made in the bad commit. To do this use the command

```
git revert <name of bad commit>
```

19. What is SubGit?

SubGit is a tool for SVN to Git migration. It can create a writable Git mirror of a local or remote Subversion repository and use both Subversion and Git as long as you like.

Now you can also include some advantages like you can do a fast one-time import from Subversion to Git or use SubGit within Atlassian Bitbucket Server. We can use SubGit to create a bi-directional Git-SVN mirror of an existing Subversion repository. You can push to Git or commit to Subversion as per your convenience. Synchronization will be done by SubGit.

20. What is the difference between git pull and git fetch?

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

Git pull = git fetch + git merge

21. What is ‘staging area’ or ‘index’ in Git?

That before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’. From the diagram it is evident that every change is first verified in the staging area I have termed it as “stage file” and then that change is committed to the repository.