**1) What is GIT?**

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and eﬃciency.

**2) What is a repository in GIT?**

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**3) What is the command you can use to write a commit message?**

The command that is used to write a commit message is “git commit –a”. The –a on the command line instructs git to commit the new content of all tracked ﬁles that have been modiﬁed. You can use “git add<ﬁle>” before git commit –a if new ﬁles need to be committed for the ﬁrst time.

**4) What is the diﬀerence between GIT and SVN?**

The diﬀerence between GIT and SVN is a) Git is less preferred for handling extremely large ﬁles or frequently changing binary ﬁles while SVN can handle multiple projects stored in the same repository. b) GIT does not support ‘commits’ across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout. c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

**5) What are the advantages of using GIT?**

a) Data redundancy and replication

b) High availability

c) Only one.git directory per repository

d) Superior disk utilization and network performance

e) Collaboration friendly

f) Any sort of projects can use GIT

**6) What language is used in GIT?**

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages. 7) What is the function of ‘GIT PUSH’ in GIT? ‘GIT PUSH’ updates remote refs along with associated objects. 8) Why GIT better than Subversion? GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes. Multiple developers can checkout, and upload changes and each change can then be attributed to a speciﬁc developer.

**9) What is “Staging Area” or “Index” in GIT?**

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**10) What is GIT stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

**11) What is GIT stash drop?**

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command. It will remove the last added stash item by default, and it can also remove a speciﬁc item if you include as an argument.

**12) How will you know in GIT if a branch has been already merged into master?**

Git branch—merged lists the branches that have been merged into the current branch Git branch—-no merged lists the branches that have not been merged

**13) What is the function of git clone?**

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

**14) What is the function of ‘git conﬁg’?**

The ‘git conﬁg’ command is a convenient way to set conﬁguration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be deﬁned through this command.

**15) What does commit object contain?**

a) A set of ﬁles, representing the state of a project at a given point of time

b) Reference to parent commit objects c) An SHAI name, a 40 character string that uniquely identiﬁes the commit object.

**16) How can you create a repository in Git?**

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

**17) What is ‘head’ in git and how many heads can be created in a repository?**

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

**18) What is the purpose of branching in GIT?**

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

**19) What is the common branching pattern in GIT?**

The common way of creating branch in GIT is to maintain one as “Main“ branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

**20) How can you bring a new feature in the main branch?**

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

**21) What is a ‘conﬂict’ in git?**

A ‘conﬂict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**22) How can conﬂict in git resolved?**

To resolve the conﬂict in git, edit the ﬁles to ﬁx the conﬂicting changes and then add the resolved ﬁles by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**23) To delete a branch what is the command that is used?**

Once your development branch is merged into the main branch, you don’t need development branch. To delete a branch use, the command “git branch –d [head]”.

**24) What is another option for merging in git?**

“Rebasing” is an alternative to merging in git.

**25) What is the syntax for “Rebasing” in Git?**

The syntax used for rebase is “git rebase [new-commit] “

**26) What is the diﬀerence between ‘git remote’ and ‘git clone’?**

‘git remote add’ just creates an entry in your git conﬁg that speciﬁes a name for a particular URL. While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

27) What is GIT version control?

With the help of GIT version control, you can track the history of a collection of ﬁles and includes the functionality to revert the collection of ﬁles to another version. Each version captures a snapshot of the ﬁle system at a certain point of time. A collection of ﬁles and their complete history are stored in a repository.

**28) Mention some of the best graphical GIT client for LINUX?**

Some of the best GIT client for LINUX is a) Git Cola b) Git-g c) Smart git d) Giggle e) Git GUI f) qGit

**29) What is Subgit? Why to use Subgit?**

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration. Subgit is a solution for a company wide migration from SVN to Git that is: a) It is much better than git-svn b) No requirement to change the infrastructure that is already placed c) Allows to use all git and all sub-version features d) Provides genuine stress –free migration experience.

**30) What is the function of ‘git diﬀ ’ in git?**

‘git diﬀ ’ shows the changes between commits, commit and working tree etc.

**31) What is ‘git status’ is used for?**

As ‘Git Status’ shows you the diﬀerence between the working directory and the index, it is helpful in understanding a git more comprehensively.

**32) What is the diﬀerence between the ‘git diﬀ ’and ‘git status’? ‘**

git diﬀ’ is similar to ‘git status’, but it shows the diﬀerences between various commits and also between the working directory and index.

**33) What is the function of ‘git checkout’ in git?**

A ‘git checkout’ command is used to update directories or speciﬁc ﬁles in your working tree with those from another branch without merging it in the whole branch.

**34) What is the function of ‘git rm’?**

To remove the ﬁle from the staging area and also oﬀ your disk ‘git rm’ is used.

**35) What is the function of ‘git stash apply’?**

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

**36) What is the use of ‘git log’?**

To ﬁnd speciﬁc commits in your project history- by author, date, content or history ‘git log’ is used.

**37) What is ‘git add’ is used for? ‘**

git add’ adds ﬁle changes in your existing directory to your index.

**38) What is the function of ‘git reset’?**

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

**39) 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.

**40) How git instaweb is used?**

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

**41) What does ‘hooks’ consist of in git?**

This directory consists of Shell scripts which are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

**42) Explain what is commit message?**

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modiﬁcations made in commits.

**43) How can you ﬁx a broken commit?**

To ﬁx any broken commit, you will use the command “git commit—amend”. By running this command, you can ﬁx the broken commit message in the editor.

**44) Why is it advisable to create an additional commit rather than amending an existing commit?** There are couple of reason

a) The amend operation will destroy the state that was previously saved in a commit. If it’s just the commit message being changed then that’s not an issue. But if the contents are being amended then chances of eliminating something important remains more. b) Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

**45. What is bare repositary ?**

A “bare” repository in Git just contains the version control information 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, where as working directory consist of:

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

46. **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

47. **How do you find a list of files that has changed in a particular commit?**

For this answer instead of just telling the command, explain what exactly this command will do.

To get a list files that has changed in a particular commit use the below command:

**git diff-tree -r {hash}**

Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory names only.

You can also include the below mentioned point, although it is totally optional but will help in impressing the interviewer.

The output will also include some extra information, which can be easily suppressed by including two flags:

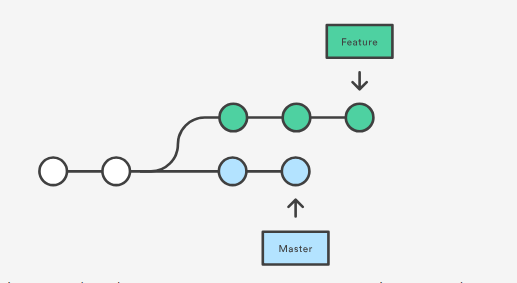
**git diff-tree –no-commit-id –name-only -r {hash}**

Here –no-commit-id will suppress the commit hashes from appearing in the output, and –name-only will only print the file names, instead of their paths.

48. GIT merge Example

git checkout feature

git merge master



git checkout feature

git rebase master

This moves the entire feature branch to begin on the tip of the master branch, effectively incorporating all of the new commits in master. But, instead of using a merge commit, rebasing re-writesthe project history by creating brand new commits for each commit in the original branch.

he major benefit of rebasing is that you get a much cleaner project history. First, it eliminates the unnecessary merge commits required by git merge. Second, as you can see in the above diagram, rebasing also results in a perfectly linear project history

GITHUB :

Git fork and pull requeste are related to GITHUB feature and nothing can bre modified in local git repositary

Git fork is similar to copyting the remote repositary into our github repositary and contribute our ideas.cchanges to files and directories in forked repo. Then we have to raise pull request to the orinal to reflect the changes that are made.