1)What is Git?

Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage 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 on his hard drive so when there is a server outage all you need for recovery is one of your teammate’s local Git repository.There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are commiting changes “Remote repository”

2) What is the command to write a commit message in Git?

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.

3) 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.

4) In Git how do you revert a commit that has already been pushed and made public?

There can be two answers to 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 push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use  
git commit -m “commit message”

Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command  
git revert <name of bad commit>

Q5. 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

Q6. What is ‘staging area’ or ‘index’ in Git?

For this answer try to explain the below diagram as you can see:

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.

7.What is Git stash?

According to me you should first explain the need for Git stash.

Often, when you’ve been working on part of your project, things are in a messy state and you want to switch branches for sometime to work on something else. The problem is, you don’t want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is Git stash.

Now explain what is Git stash.

Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.

Q8. What is Git stash drop?

Begin this answer by saying for what purpose we use Git ‘stash drop’.

Git ‘stash drop’ command is used to remove the stashed item. It will remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

Now give an example.

If you want to remove a particular stash item from the list of stashed items you can use the below commands:

git stash list: It will display the list of stashed items like:  
stash@{0}: WIP on master: 049d078 added the index file  
stash@{1}: WIP on master: c264051 Revert “added file\_size”  
stash@{2}: WIP on master: 21d80a5 added number to log

If you want to remove an item named stash@{0} use command git stash drop stash@{0}.

Q9. 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.

Q10. What is the function of ‘git config’?

First tell why we need ‘git config‘.

Git uses your username to associate commits with an identity. The git config command can be used to change your Git configuration, including your username.

Now explain with an example.

Suppose you want to give a username and email id to associate commit with an identity so that you can know who has made a particular commit. For that I will use:

git config –global user.name “Your Name”: This command will add username.  
git config –global user.email “Your E-mail Address”: This command will add email id.

Q11. What does commit object contains?

Commit object contains the following components, you should mention all the three points present below:

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

Reference to parent commit objects

An SHAI name, a 40 character string that uniquely identifies the commit object.

Q12. How can you create a repository in Git?

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

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

Q13. What is Git bisect? How can you use it to determine the source of a (regression) bug?

I will suggest you to first give a small definition of Git bisect.

Git bisect is used to find the commit that introduced a bug by using binary search. Command for Git bisect is  
git bisect <subcommand> <options>

Now since you have mentioned the command above explain them what this command will do.

This command uses a binary search algorithm to find which commit in your project’s history introduced a bug. You use it by first telling it a “bad” commit that is known to contain the bug, and a “good” commit that is known to be before the bug was introduced. Then Git bisect picks a commit between those two endpoints and asks you whether the selected commit is “good” or “bad”. It continues narrowing down the range until it finds the exact commit that introduced the change.

14.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

15. What are the advantages of using GIT?

Distributed System: GIT is a Distributed Version Control System (DVCS). So you can keep your private work in adaptation control yet totally escaped others. You can work disconnected too.

⦁ Flexible Workflow:GIT enables you to make your own work process. You can utilize the procedure that is appropriate for your venture. You can go for brought together or ace slave or some other work process.

⦁ Fast: GIT is quick when contrasted with other form control frameworks.

⦁ Data Integrity: Since GIT utilizes SHA1, information isn’t less demanding to degenerate.

⦁ Free: It is free for individual utilize. Such huge numbers of beginners utilize it for their underlying activities. It likewise works exceptionally well with substantial size task.

⦁ Collaboration: GIT is anything but difficult to use for ventures in which joint effort is required. Numerous prevalent open source programming over the globe utilize GIT

16. 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.

17. 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.

18.  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.

AHEAD is a reference to the present looked at conferring.

It is a representative reference to the branch that we have looked at.

At any given time, one head is chosen as the ‘present head’ this head is otherwise called HEAD (dependably in capitalized).

19)   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.

20)   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.

21)   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”.

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 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.

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

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

29)   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.

30)   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.

31)   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.

32)   What is the function of ‘git rm’?

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

33)   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.

34)   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.

35)   What is ‘git add’ is used for?

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

36)   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.

37)   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.

38)   How git instaweb is used?

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

39)   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.

40)   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.

41)   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.

42)   Name a few Git repository hosting services

Pikacode

Visual Studio Online

GitHub

GitEnterprise

SourceForge.net

How do you cherry-pick a merge commit?

*43)Cherry-pick* uses a diff to find the difference between branches.

As a merge commit belongs to a different branch, it has two parents and two changesets.

For example, if you have merge commt ref 63ad84c, you have to specify -m and use parent 1 as a base:

# git checkout release\_branch

# git cherry-pick -m 1 63ad84c

**44)What is Git fork? What is difference between fork and branch? How to create tag?**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

A fork is really a Github (not Git) construct to store a clone of the repo in your user account. As a clone, it will contain all the branches in the main repo at the time you made the fork.

***Create Tag:***

– Click the releases link on our repository page.  
– Click on Create a new release or Draft a new release.  
– Fill out the form fields, then click Publish release at the bottom.  
– After you create your tag on GitHub, you might want to fetch it into your local repository too: git fetch.

**45. How to rebase master in git? Difference between rebase and merge. How to squash or fixup commits?**

Rebasing is the process of moving a branch to a new base commit.

The golden rule of git rebase is to never use it on public branches. … The only way to synchronize the two master branches is to merge them back together, resulting in an extra merge commit and two sets of commits that contain the same changes.

46**Name some GIT commands and also explain their functions?**

* **GIT diff** – It shows the changes between commits, commits and working tree.
* **GIT status** – It shows the difference between working directories and index.
* **GIT stash applies** – It is used to bring back the saved changes on the working directory.
* **GIT rm** – It removes the files from the staging area and also of the disk.
* **GIT log** – It is used to find the specific commit in the history.
* **GIT add** – It adds file changes in the existing directory to the index.
* **GIT reset** – It is used to reset the index and as well as the working directory to the state of the last commit.
* **GIT checkout** – It is used to update the directories of the working tree with those from another branch without merging.
* **GIT Is tree** – It represents a tree object including the mode and the name of each item.
* **GIT**instaweb – It automatically directs a web browser and runs the web server with an interface into your local repository.

47. **What is the purpose of branching and how many branching strategies can you apply?**

The branching in GIT is done to allow the user to create their own branch and toggle between those branches. It will allow the user to go to the previous work keeping the current work intact.

**Branching Strategies:**

* **Feature Branching** – It keeps all the changes of a specific feature in a branch and when the feature is completely tested and validated it is merged into the master.
* **Task branching** – It is included in its own branch with the task key included in the branch name.
* **Release Branching** – When the developed branch acquires enough features for a release, the user can clone that branch to form a releasing branch.

**Q48). Describe Feature branching strategy in Git.**

A feature branch type of Git model keeps the entire set of modifications for any specific characteristic within the branch. Only when this specific characteristic is entirely tested and verified by several automated tests, then this branch is merged into Git’s master branch.

**Q49). Describe Task branching strategy in Git.**

In this type of Git model, each and every task is first executed on its very own branch along with the task key duly contained in the branch name itself. It is quite simple to observe as to which code is going to execute which task.