GIT INTERVIEW QUESTIONS

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 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.

3)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.

4) 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 is private to git.

5) What are the advantages of using GIT?

a) Data redundancy and replication

b) High availability

c) Only one. Its directory per repository

d) Superior disk utilization and network performance

e) Collaboration friendly

f) Any sort of projects can use GIT

6)What is the function of 'GIT PUSH' in GIT?

'GIT PUSH' updates remote refs along with associated objects. 7) 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.

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 check out, 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) 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".

16) 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.

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 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) A SHAI name, a 40-character string that uniquely identifies the commit object.

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) 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 statuses, 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 insta web is used?

'Git Insta web' 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 repository' in GIT?

To co-ordinate with the distributed development and developers' team, especially when you are working on a project from multiple computers 'Bare Repository' is used. A bare repository comprises of a version history of your code .

46) Name a few Git repository hosting services?

Pika code
Visual Studio Online
GitHub
Gi Enterprise
SourceForge.net

47) What is the difference between Git and GitHub?

Git is a revision control system, a tool to manage your source code history. GitHub is a hosting

service for Git repositories.

GitHub is a website where you can upload a copy of your Git repository. It is a Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

48) How do you rate GIT in terms of speed?

Git is fast. Speed and performance have been a primary design goal of the Git from the start. With Git, nearly all operations are performed locally, giving it a huge speed advantage on centralized systems that constantly have to communicate with a server somewhere.

Git was built to work on the Linux kernel, meaning that it has had to effectively handle large repositories from day one. Git is written in C, reducing the overhead of runtimes associated with higher-level languages.

49) 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>

50) 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

51) How do you find a list of files that has changed in a particular commit?

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.
The output will also include some extra information, which can be easily suppressed by including a

couple of flags:

git diff-tree --no-commit-id --name-only -r {hash}
Here *--no-commit-id* will supress the commit hashes from appearing in the output, and *-- name-only* will only print the file names, instead of their paths.

52) How do you setup a script to run every time a repository receives new

commits through push?

To configure a script to run every time a repository receives new commits through push, one needs to define either a pre-receive, update, or a post-receive hook depending on when exactly the script needs to be triggered.

Pre-receive hook in the destination repository is invoked when commits are pushed to it. Any script bound to this hook will be executed before any references are updated. This is a useful hook to run scripts that help enforce development policies.

Update hook works in a similar manner to pre-receive hook, and is also triggered before any updates are actually made. However, the update hook is called once for every commit that has been pushed to the destination repository.

Finally, post-receive hook in the repository is invoked after the updates have been accepted into the destination repository. This is an ideal place to configure simple deployment scripts, invoke some continuous integration systems, dispatch notification emails to repository maintainers, etc. Hooks are local to every Git repository and are not versioned. Scripts can either be created within the hooks directory inside the ". git" directory, or they can be created elsewhere and links to those scripts can be placed within the directory.

53)What is git bisect? How can you use it to determine the source of a

(regression) bug?

Git provides a rather efficient mechanism to find bad commits. Instead of making the user try out every single commit to find out the first one that introduced some particular issue into the code, git bisect allows the user to perform a sort of binary search on the entire history of a repository. By issuing the command git bisect start, the repository enters bisect mode. After this, all you have to do is **identify a bad and a good commit:**

```
bisect bad
```

```
 git # marks the current version as bad
```

```
git bisect good {
```

```
or tag}
```

```
hash # marks the given hash or tag as good,
```

```
ideally of some earlier commit
```

Once this is done, Git will then have a range of commits that it needs to explore. At every step, it will checkout a certain commit from this range, and require you to identify it as good or bad. After which the range will be effectively halved, and the whole search will require a lot less number of steps than the actual

number of commits involved in the range. Once the first bad commit has been found, or the bisect mode needs to be ended, the following command can be used to exit the mode and reset the **bisection state:**

```
bisect reset
```

git

54)How do you cherry-pick a merge commit?

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 commit ref 63ad84c, you have to specify -m and use parent 1 as a base:

```
git_checkout release branch
```

```
git_cherry-pick -m 1 63ad84c
```

55) One of your teammates accidentally deleted a branch, and has already pushed the changes to the central git repo. There are no other git repos, and none of your other teammates had a local copy. How would you recover this branch?

Check out the latest commit to this branch in the reflog, and then check it out as a new branch.