GIt

1. **What is GIT and What are it's advantages?**
2. **What is the relationship between GIT and SCM tools?**
3. **Can you explain about Branching and Merging in GIT?**
4. **How do you rate GIT in terms of speed?**
5. **What is a pull in git?**
6. **What does git commit a?**
7. **Why do you use GIT?**
8. **What is the purpose of Git?**
9. **What do you mean by git add?**
10. **What is the difference between Git and Github?**
11. **What does git pull rebase do?**
12. **What does git pull origin master do?**
13. **What is the difference between SVN and Git?**
14. **What is a stash in git?**
15. **What is git pull origin?**

**1) What is GIT and What are it's advantages?**

A) Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

**2) What is the relationship between GIT and SCM tools?**

A) SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

**3) Can you explain about Branching and Merging in GIT?**

A) The Git feature that really makes it stand apart from nearly every other SCM out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

**4) How do you rate GIT in terms of speed?**

A) Git is fast. Speed and performance has 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.

**5) What is a pull in git?**

A) git-pull - Fetch from and integrate with another repository or a local branch

SYNOPSIS: git pull [options] [ […?]]

In its default mode, git pull is shorthand for git fetch followed by git merge FETCH\_HEAD. More precisely, git pull runs git fetch with the given parameters and calls git merge to merge the retrieved branch heads into the current branch. should be the name of a remote repository as passed to git-fetch

**6) What does git commit a?**

A) Basically git commit "records changes to the repository" while git push "updates remote refs along with associated objects". So the first one is used in connection with your local repository, while the latter one is used to interact with a remote repository.

**7) Why do you use GIT?**

A) Git is a version control system (VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development but it can be used to keep track of changes in any set of files.

**8) What is the purpose of Git?**

A) The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository.

**9) What do you mean by git add?**

A) git add . adds all modified and new (untracked) files in the current directory and all subdirectories to the staging area (a.k.a. the index), thus preparing them to be included in the next git commit. Any files matching the patterns in the .gitignore file will be ignored by git add.

**10) What is the difference between Git and Github?**

A) 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.

**11) What does git pull rebase do?**

A) git pull --rebase is allows you to later squash your commits to a few (or one) commits. If you have merges in your (unpushed) history, it is not so easy to do a git rebase later one.

**12) What does git pull origin master do?**

A) git pull origin master pulls the master branch from the remote called origin into your current branch. It only affects your current branch, not your local master branch.

**13) What is the difference between SVN and Git?**

A) In short, svn is a Centralized Revision Control System, and git is a Distributed Revision Control System (DVCS).

**14) What is a stash in git?**

A) Stashing takes the dirty state of 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.

**15) What is git pull origin?**

A) pull is a fetch and a merge. \* `git pull origin master` fetches commits from the master branch of the origin remote (into the local origin/master branch), and then it merges origin/master into the branch you currently have checked out.

**16)Define 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.

**17)Define ‘bare repository’ in Git?**

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

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

**18)What are the different ways you can refer to a commit?**

In Git each commit is given a unique hash. These hashes can be used to identify the corresponding commits in various scenarios (such as while trying to checkout a particular state of the code using the git checkout {hash} command).

Additionally, Git also maintains a number of aliases to certain commits, known as refs. Also, every tag that you create in the repository effectively becomes a ref (and that is exactly why you can use tags instead of commit hashes in various git commands). Git also maintains a number of special aliases that change based on the state of the repository, such as HEAD, FETCH\_HEAD, MERGE\_HEAD, etc.

Git also allows commits to be referred as relative to one another. For example, HEAD~1 refers to the commit parent to HEAD, HEAD~2 refers to the grandparent of HEAD, and so on. In case of merge commits, where the commit has two parents, ^ can be used to select one of the two parents, e.g. HEAD^2 can be used to follow the second parent.

And finally, refspecs. These are used to map local and remote branches together. However, these can be used to refer to commits that reside on remote branches allowing one to control and manipulate them from a local Git environment.

**19)What is a conflict in git and how can it be resolved?**

A conflict arises when more than one commit that has to be merged has some change in the same place or same line of code. Git will not be able to predict which change should take precedence. This is a git conflict.

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 merge, so it sets the parents of the commit correctly.

**20)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.

**21)What is SubGit?**

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

**22)What is the HEAD in GIT ?**

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).

**23)What are Git Design objectives?**

Distributed workflow (decentralised)

Easy merging (merge deemed more frequent than commit)

Integrity (protection against accidental/malicious corruptions)

Speed & scalability

**24)What is Version Control with Git?**

Version control is better than mailing files back and forth because:

* It’s is not impossible to coincidentally overwrite or overlook someone’s changes: whenever there’s a conflict between one person’s work and another’s, the version control system automatically notifies users.
* If people are having some questions to ask, they will maintain the records what changes they have made.
* Nothing that is committed to version control is ever lost. it’s always possible to go back in time to know exactly who wrote what on a particular day, or what version of a program was used to generate a particular set of results. This means it can be used like the undo feature in an editor, and since all previous versions of files are saved.

Git is one of many version control systems. It is more complex than some alternatives, but it is widely used, both because it’s easy to set up and because of a hosting site called GitHub, which we will get to later.

**25)What are the main benefits of 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

Top of Form

GIT

**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?**

[](https://career.guru99.com/wp-content/uploads/2014/04/Git.jpg)

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 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**

* Pikacode
* Visual Studio Online
* GitHub
* GitEnterprise
* SourceForge.net