GIT Interview Questions & Answers

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 Distributed Control System?

We work in our local machine and later we transfer the code to Centralized repository (GitHub). We don't need to connect to centralized repository to work.

3. What is GIT version control?

- GIT version control allows you to track the history of a collection of files (code files).
- It supports creating different versions of file collection. Each version captures a snapshot of the files at a certain point of time and You can revert the collection of files using the snapshot. (You can develop the code in different versions of java. and you can merge in Git)
- VCS allows you to switch between these versions. These versions are stored in a specific place, typically called as repository. (You can switch between different versions of java in between development process)

4. What is difference between SVN and Git?

SVN	GIT
SVN is centralized repository, that means directly we involved in the centralized repository.	Git is distributed repository, first we are working in our laptop after that we are transferring the code from our laptop to centralized repository. Git have three phases the phases are work space, staging/index, local repo.
We working on SVN means if we are facing any networking issue we can't work on SVN because of we are directly involve into the centralized repository.	In git we are doing in local systems only so no need to internet connection, when pushing the code from our system to centralized repository at that time we need network connection. Without network also, we can do some work.
Developed directly interact with the centralized repository.	Developers not directly interact with the Centralized repository.

5. What is a repository in GIT?

A Git repository contains the history of a files.

6. How can you create a local repository in Git?

By using # git init command create a local repository.

7. What is 'bare repository' in GIT?

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.

8. How to configure GitHub repository locally?

git config --global user.name "user_name"

git config --global user.email "user_email"

9. How to Create Alias to git commands

git config --global alias.lo "log --oneline" -----> To create an Alias to Command

git config --global --unset alias.lo -----> To Remove an Alias

git config --global --unset user.name -----> to remove username

10. What is the git clone?

To download an existing repository from Centralized (Github) to local system.

git clone <url>

11. What is 'git add'?

To add files from work area to Index/staging/cache area.

git add <file_name1> <file_name2>

12. What is Staging Area?

staging area means "holding area". Before the commits, it can be formatted and reviewed in an intermediate area known as staging or Index Area.

13. What is the use of 'git log'?

To see the commits. Also, we can find specific commits in your project history- by author, date, content or history.

git log -----> To show the Git Commits

git log -5 -----> To show Recent 5 Commits

git log --oneline -----> To Display the each commit in one line

git log --since=2018-01-21

git log --until=2018-03-18

git log --author="user_name"

git log --grep="Index"

git log --oneline --author="user_name"

14. How can we add modified/updated/edited files to the staging area and commit then at the same time?

git commit -a -m "Do Something once more"

15. How to edit an incorrect commit message in Git? Or How can you fix a broken commit?

git commit --amend -m "This is your new Git Message"

16. How to get back a commit to staging area?

git reset --soft <previous_commit id>

17. How to get back a file from staging area to working area?

git reset head <file_name>

18. How to get back a commit to work area?

git reset --mixed <previous commit id>

19. What is git reset?

Reset the current HEAD state to specific state.

20. 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.

21. What is .gitignore file?

Keep the files names in .gitignore then that files not add and commit, just skip that files while adding and committing.

22. How to see the difference between 2 commits?

git diff <commit_id1>..<commit_id2>

23. when file have staging area or file have committed if file is deleted in local repository unfortunately how to get back that file to staging area?

git checkout --<file_name>

24. How to create a branch?

git branch <branch_name>

25. How to checkout to branch?

git checkout <branch_name>

26. How to create branch while checkout?

git checkout -b <branch_name>

27. How do you rename the local branch?

git branch -m <old_branch_name> <new_branch_name>

28. How to see the branch list?

git branch

29. How to see the remote branch list?

git branch -r

Or

git remote show origin

30. How to see the local and remote branch list?

git branch -a

31. How to delete a branch?

git branch -d <branch_name>

Or

git branch -D <branch_name>

32. How to delete a Remote Branch?

git push origin -d <branch_name>

33. How to see the difference between 2 branches

git diff <branch1>..<branch2 >

34. What is git push?

git push is to push commits from your local repository to a remote repository.

35. How do you push the files to master branch in remote repo?

#git push (you must be in master branch)

36. How do you push files from local to particular branch in remote repo?

#git push origin <branch_name>

(or)

#git push --set-upstream <branch_name>

37. How to push new branch and its data to remote repository?

#git push <github_repository_path> <branch_name>

(or)

#git push --set-upstream <branch_name>

38. What is git pull?

Git pull downloads and merges a 'branch data' from remote repository to local repository. It may also lead to 'merge conflicts' if your local changes are not yet committed. Use 'git stash' command to Hide your local changes before git pull.

git pull (git fetch + git merge.)

39. How do you pull a file from particular remote branch?

git pull origin <branch_name>

40. How do you download a remote branch to local without merge?

git fetch origin <branch_name>

git checkout <downloaded_branchname>

41. What is git Fetch?

git fetch is only downloads new data from a remote repository, but it doesn't integrate any of the downloaded data into your working files. All it does is provide a view of this data.

git fetch <branch_name>

git fetch origin <branch_name>

42. What is difference between git clone & git pull?

- If you want to download whole existing repository than use Git Clone.
- If you have already repository but you want to take new updates of existing repository than use git pull command.

43. What is git merge?

Git merge is used to combine two branches.

git merge <branch_name>

Note: you should be in target branch. Then run the command

44. What is git conflict? What is the scenario you will get git conflict error?

For example, if you and another person both edited the same file on the same lines in different branches of the same Git repository, you'll get a merge conflict error when you try to merge these branches. You must resolve this merge conflict with a new commit before you can merge these branches.

45. How do you resolve merge conflict?

Will inform the developers regarding this merge conflict. They will change the code and inform us. edit the files to fix the conflicting changes and then add & commit.

46. How do you skip from merge conflict?

#git merge --abort

47. What is the function of 'git rm'?

To remove the file from the work area/staging area and also from your disk 'git rm' is used. You can revert a deleted file.

if it is deleted using 'git rm'. If you deleted a file 'rm' command then you can't get it.

48. How will you know in GIT if a branch has been already merged into master?

git branch -merged It lists the branches that have been merged into the current branch.

git branch -no-merged It lists the branches that have not been merged.

49. What is branching? What is the purpose of branching in GIT?

Git supports branching which means that you can work on different versions of your collection of files. A branch allows the user to switch between these versions so that he can work on different changes independently from each other.

50. What is the criteria u merge two branches?

We have developed one module in one branch and another module in another branch. After the development, based on the requirement we do merge these two branches.

Or One branch is development branch, another branch is test branch.

51. Describe branching strategy you have used?

Feature branching

A feature branch model keeps all of the changes for a particular feature inside of a branch. When the feature is fully tested and validated by automated tests, the branch is then merged into master.

Task branching

In this model each task is implemented on its own branch with the task key included in the branch name. It is easy to see which code implements which task, just look for the task key in the branch name.

Release branching

Once the develop branch has acquired enough features for a release, you can clone that branch to form a Release branch. Creating this branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it is ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into develop branch, which may have progressed since the release was initiated.

52. What is git stash?

Stashing takes the Temporary stored state of your working directory.

git stash save "<message>" ------> to store the data into stash

git stash list ------> to see the stash list

git stash apply <stash#> ------> to copy the data into branches

git stash pop <stash#> ------> to move the data into branches

git stash drop <stash#> ------> to delete the particular stash

git stash clear ------> delete the entire stash list

53. When we use git Stash?

- If you are checking out from one branch to another branch but you have uncommitted file that you don't want to move then keep that file in stash area.
- When you are merging two branches and you don't want some files to merge, then we move that files to stash area.
- When you are pulling (fetch + merge) a branch/file and you don't want some files to merge, then we move that files to stash area.

54. What is another option for merging in git?

git rebasing command is an alternative to merging in git.

55. What is difference between git merge and git rebase?

- git merge applies all unique commits from branch A into branch B in one commit with final result.
- git rebase gets all unique commits from both branches and applies them one by one.
- git merge doesn’t rewrite commit history, just adds one new commit
- git rebase rewrites commit history but doesn’t create extra commit for merging

56. How do you undo the last commit?

git revert <commit_id>

57. How to Change the URL for a remote Git repository?

git remote set-url origin git://this.is.new.url

58. What is pull request?

Take some changes from a particular branch and bring them into another branch.

59. Why GIT better than Subversion (SVN)?

Git is an opensource version control system; it will allow you to run ‘version' of a project. Multiple developers can check out, and upload changes and each change can then be attributed to a specific developer.

60. How to Lock a branch? why we need to lock a branch?

- On GitHub, navigate to the main page of the repository.
- Under your repository name, click **Settings**.
- In the left menu, click **Branches**.
- Select the branch you want to mark protected using the drop-down menu.
- Select **Protect this branch**.

61. How to delete Repository in GitHub?

- On GitHub, navigate to the main page of the repository.
- Under your repository name, click **Settings**.
- Scroll to the bottom of the page and you will find **Delete this repository** button
- When you click on that button, another pop up will appear, here you need to type in the name of your repository name and click on the button bellow which says: I understand the consequences, delete the repository.

62. how to give an access to a specific person to repository?

You can invite users to become collaborators to your personal repository.

- Under your repository name, click **Settings.**
- In the left sidebar, click **Collaborators**.
- Under "Collaborators", start typing the collaborator's username.
- Select the collaborator's username from the drop-down menu.
- Click **Add collaborator**.
- The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.