**Version Control System (VCS) Interview Questions**

**Q1. What is Version control?**

It is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control systems consist of a central shared repository where teammates can commit changes to a file or set of file.

**Q2. What are the benefits of using version control?**

* Revert files back to a previous state.
* Revert the entire project back to a previous state.
* Compare changes over time.
* See who last modified something that might be causing a problem.
* Who introduced an issue and when.

**Q3. 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.
* 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 committing changes “Remote repository”.

**Q4. Command to initialize git repository?**

Git init

**Q5. Git command to configure author?**

* git config –global user.name”name”
* git config –global user.email”E-mail”

**Q5. How do you create working copy of local repository?**

git clone /path/to/repository

**Q6. How do you create working copy of remote repository?**

git clone <https://user@example.com/path/to/my-project.git> <directory>

**Q7. How do you add files to staging area?**

Git add.-🡪to add all files

Git add <file name> 🡪 to add specific file

**Q8. How do you commit changes to local repository?**

Git commit -m “commit message”

**Q9. How do you connect local repository to remote repository?**

Git remote add origin <server>

**Q10. How do you commit changes to remote repository?**

Git push origin <master/branch name>

**Q11. In Git how do you revert a commit that has already been pushed and made public?**

1. Remove or fix the bad file in a new commit and push it to the remote repository.  
   **git commit -m “commit message”**
2. Create a new commit that undoes all changes that were made in the bad commit.  
   **git revert <name of bad commit>**

**Q12. How do you squash last N commits into a single commit?**

If you want to write the new commit message from scratch

**git reset –soft HEAD~N &&git commit**

**Q13. What is Git rebase?**

Git rebase is a command which will merge another branch into the branch where you are currently working, and move all of the local commits that are ahead of the rebased branch to the top of the history on that branch.

**Q14. How can it be used to resolve conflicts in a feature branch before merge?**

If a feature branch was created from master, and since then the master branch has received new commits, Git rebase can be used to move the feature branch to the tip of master.

**Q15. How do you find a list of files that has changed in a particular commit?**

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

**Q16. How will you know in Git if a branch has already been merged into master?**

1. **git branch –merged** lists the branches that have been merged.
2. **git branch –no-merged** lists the branches that have not been merged.

**Q17. 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’**.**

**Q18. 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.

**Q19. 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.

**Q20. 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

**Q21. 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.

**Q22. 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.

**Q23. 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.

**Q24. 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.

**Q25. 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]”.

**Q25. 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

**Q26. 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

**Q27. 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.

**Q28. What is the function of ‘git checkout’ in git?**

Git-checkout - Switch branches or restore working tree files

**Q29. What is the function of ‘git rm’?**

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

**Q30. 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.

**Q31. 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.

**Q32. 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.

**Q33. What is ‘bare repository’ in GIT?**

To co-ordinate with the distributed development and developer’s 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.

**Q34. Name a few Git repository hosting services**

* GitHub
* Pikacode
* Visual Studio Online