What's the git command that downloads your repository from GitHub to your computer?

Top of Form

git push

git fork

**git clone**

git commit

Bottom of Form

How do you create a copy of a lab under your own GitHub account so that you can solve the lab?

Top of Form

**Forking it via the GitHub interface.**

git fork

git clone

git pull-request

Bottom of Form

What's the opposite of git clone, instead of downloading your code from GitHub, uploads your changes and code back to GitHub?

Top of Form

**git push**

git add

git upload

git status

Bottom of Form

How do you check the state of your local git repository since your last commit?

Top of Form

git check

**git status**

git commit

git diff

Bottom of Form

How do you stage files for a commit?

Top of Form

git stage

git commit

**git add**

git reset

Bottom of Form

How do you save the current state of your code into the git version control?

Top of Form

**By committing the staged changes with git commit**

By adding all changes and staging them with git stage

By adding all changes and staging them with git add

By creating a new commit with git init

Bottom of Form

What's a shortcut to staging all the changes you have?

Top of Form

git commit add .

git commit .

**git add .**

git push -am "Message"

Bottom of Form

How do you supply a commit message to a commit?

Top of Form

git message "I'm coding"

git add "I'm coding"

git commit "I'm coding"

**git commit -m "I'm coding"**

Bottom of Form

What is the correct commit syntax for all changes with a message?

Top of Form

git message -am "I'm coding"

git add -a "I'm coding"

git commit -a "I'm coding"

**git commit -am "I'm coding"**

**Bottom of Form**

How do you submit a solution to Learn? (select all that apply)

Top of Form

git submit

git pull-request

**By creating a Pull Request through the GitHub interface**

**learn submit**

git commit -am "Done with Lab"

Bottom of Form

[3. \_\_\_\_\_\_\_\_\_\_\_\_ command is useful for getting a high-level overview of the project history.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled3)

 git log --oneline

 git reset --hard

 git log --author="<pattern>"

 git rebase <base>

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git log --oneline command is useful for getting a high-level overview of the project history**

[4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ removes untracked files from your working directory.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled4)

git commit

 git clean -f <path>

 git clean

*  git reset

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git clean removes untracked files from your working directory.**

[5. Which command creates an empty Git repository in the specified directory?](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled5)

*  git reset <file>
*  git log <since>..<until>
*  git init <directory>
*  git init --bare <directory>

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git init <directory> creates an empty Git repository in the specified directory?**

[6. Command to download all the objects and references from a specified repository](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled6)

*  git config --list
*  git help <command>
*  git fetch <remote>
*  git log -n <limit>

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git fetch <remote> command downloads all the objects and references from a specified repository**

[7. Git command to compare two specified branches](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled7)

*  git diff <first-branch>...<second-branch>
*  git merge <fetched-remote-name> <branch-name>
*  git blame -L <start>,<end> <filename>
*  git push <remote> --tags

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git diff <first-branch>...<second-branch> command compares two specified branches**

[8. \_\_\_\_\_\_\_\_\_\_\_\_\_ command renames the current branch to <branch>](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled8)

*  git remote rm <name>
*  git branch -m <branch>
*  git branch -D <branch> (CAPS)
*  git rebase <base>

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git branch -m <branch> command renames the current branch to <branch>**

[9. Which Git command displays the patch representing each commit.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled9)

*  git branch
*  git remote -v
*  git log -p
*  git log

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git log -p command displays the patch representing each commit.**

[10. Which of the following command line environment is used for interacting with Git ?](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled10)

*  Git Bash
*  GitHub
*  Git Boot
*  Git Lab

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**Git Bash command line environment is used for interacting with Git .**

[11. In Git, if you want to make your local repository reflect changes that have been made in a remote (tracked) repository, you should run the pull command](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled11)

*  True
*  False

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**True**

[12. If you want to make radical changes to your team’s project and don’t want to impact the rest of the team, you should implement your changes in -](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled12)

*  the root
*  a tag
*  the trunk
*  None of the above

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**None of the above**

[13. The Git clone command does which of the following?](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled13)

*  Makes a local copy of the repository
*  Creates a working directory
*  Commits a new branch
*  Both 1 & 2

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**Both 1 & 2**

[14. Which one of the following is not part of the data structure of a Git repository?](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled14)

*  Branch pointer
*  Body element
*  Commit object
*  Head pointer

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**Body element**

[15. Which of these Git client commands creates a copy of the repository and a working directory in the client’s workspace.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled15)

*  checkout
*  clone
*  import
*  update

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**clone**

[16. Git is a .................... Version Control tool.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled16)

*  Decentralized
*  Centralized

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**Decentralized**

[17. GIT belongs to the............. generation of Version Control tools](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled17)

*  2nd
*  3rd
*  4th
*  5th

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**3rd**

[18. The main objectives of Git are -](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled18)

*  speed
*  data integrity
*  support for distributed non-linear workflows
*  All of the above

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**All of the above**

[19. What language is used in Git?](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled19)

*  C
*  HTML
*  PHP
*  C++

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**C**

[20. Git command .................... used to give tags to the specified commit.](https://www.onlineinterviewquestions.com/git-mcq/" \l "collapseUnfiled20)

*  git checkout [branch name]
*  git show [commit]
*  git tag [commitID]
*  git rm [file]

[View Answer](https://www.onlineinterviewquestions.com/git-mcq/)

**git tag [commitID]**

# Multiple-Choice Questions:

1. Which of these Git client commands creates a copy of the repository and a working directory in the client’s workspace. (Choose one.)
   1. update
   2. checkout
   3. **clone**
   4. import
   5. None of the above
2. True or False? In Git, if you want to make your local repository reflect changes that have been made in a remote (tracked) repository, you should run the pull command.
   1. **True**
   2. False
3. In Git, which error would you get if you try to push master-branch changes to a remote repository, and someone else pushed changes to that same branch while you were making your changes? (Choose one.)
   1. **Rejected**

b. 404

c. 500

1. Access denied
2. 400 Bad request
3. If you want to make radical changes to your team’s project and don’t want to impact the rest of the team, you should implement your changes in …
   1. … a tag.
   2. … the trunk.
   3. … the root.
   4. **… a branch.**
   5. None of the above
4. Imagine that you just joined a development team that uses Git for version control and collaboration. To start contributing to the project, what Git operation would you most likely invoke first?
   1. checkout
   2. **clone**
   3. export
   4. revert
   5. update
5. Now, imagine that you have a local repository, but other team members have pushed changes into the remote repository. What Git operation would you use to download those changes into your working copy?
   1. checkout
   2. commit
   3. export
   4. **pull**
   5. update
6. The Git **clone** command does which of the following?
   1. Creates a working directory
   2. Makes a local copy of the repository
   3. Commits a new branch
   4. **a and b**
   5. a, b, and c
7. Which Git command changes where the HEAD pointer points and modifies the contents of the work- ing directory?
   1. **checkout**
   2. merge
   3. mv
   4. pull
   5. None of the above
8. Which one of the following is not part of the data structure of a Git repository?
   1. **Body element**
   2. Branch pointer
   3. Commit object
   4. HEAD pointer
   5. None of the above (i.e., they are all parts)

Problem:

Consider the following scenario involving Git. Alice and Bob are both working on a shared project **My- Proj** that is stored in a remote Git repository. Bob does a **clone** on the remote repository. What two things does Git create when Bob issues the **clone** command?

Next, Bob edits the **MyProj** file **foo.rb**. Then, he does a **commit** and a **push**. What does Git do when Bob issues these commands?

Next, Alice does a **clone** on **MyProj**. Then, Alice and Bob both edit **foo.rb** in parallel. foo.rb has over 100 lines of code. Alice edits a couple lines at the top of the file, and Bob edits a couple lines at the bot- tom of the file. Then, Bob does a **commit** and a **push**. Finally, Alice does a **commit** and a **push**. What does Git do when Alice issues the **push** command?

What Git commands should Alice issue next and what would the result of the command be?

Solution:

Consider the following scenario involving Git. Alice and Bob are both working on a shared project **My- Proj** that is stored in a remote Git repository. Bob does a **clone** on the remote repository. What two things does Git create when Bob issues the **clone** command?



When Bob issues the **checkout** command, Git creates a local copy of the **MyProj** repository and a working directory that contains the latest snapshot of the project files.

Next, Bob edits the **MyProj** file **foo.rb**. Then, he does an **add**, a **commit** and a **push**. What does Git do when Bob issues these commands?



The **add** commands “stages” the changes. The **commit** command updates Bob’s local repository to reflect the changes. The **push** command updates the remote repository to reflect the changes in Bob’s local repository.

Next, Alice does a **clone** on **MyProj**. Then, Alice and Bob both edit **foo.rb** in parallel. foo.rb has over 100 lines of code. Alice edits a couple lines at the top of the file, and Bob edits a couple lines at the bot- tom of the file. Then, Bob does an **add**/**commit**/**push**. Finally, Alice does a **add**/**commit**/**push**. What does Git do when Alice issues the **push** command?



When Alice issues the **push** command, Git rejects her push because the remote branch has changed since the last time she pulled from it.

What Git commands should Alice issue next and what would the result of the command be?



Alice should do a **pull** on the remote repository. That will update her current branch in her local repos- itory as well as her working directory. The update will both download the changes in the remote repos- itory and merge them into her current branch. To then upload the merged changes, she would need to do an **add**/**commit**/**push**.

# Problems:

Draw the state of the pictured repository after a Git **commit** operation (make up a hash).

After:

Before:

myﬁx

34ac2

98ca9

master

HEAD

Draw the state of the pictured repository after running the following commands.

$ git checkout master

$ git merge myfix

After:

Before:

HEAD

myﬁx

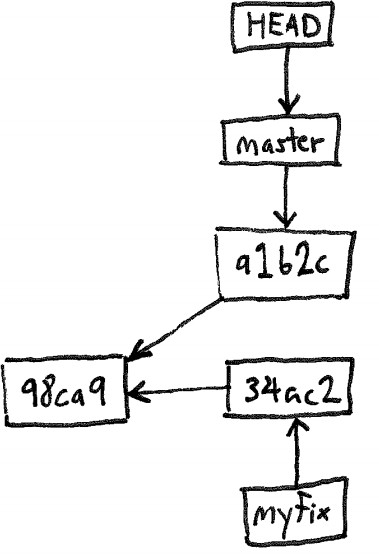
34ac2

98ca9

master

# Solutions:

Draw the state of the pictured repository after a Git **commit** operation (make up a hash).



After:

Before:

myﬁx

34ac2

98ca9

master

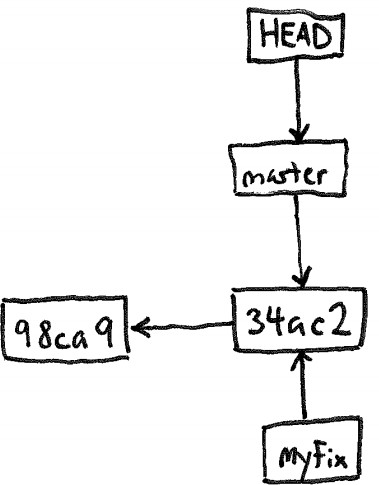
HEAD

Draw the state of the pictured repository after running the following commands.

$ git checkout master

$ git merge myfix

Problems:



After:

Before:

HEAD

myﬁx

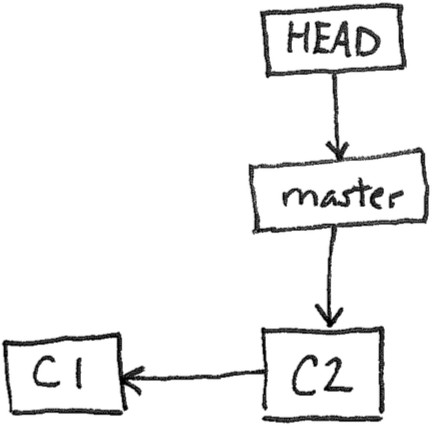
34ac2

98ca9

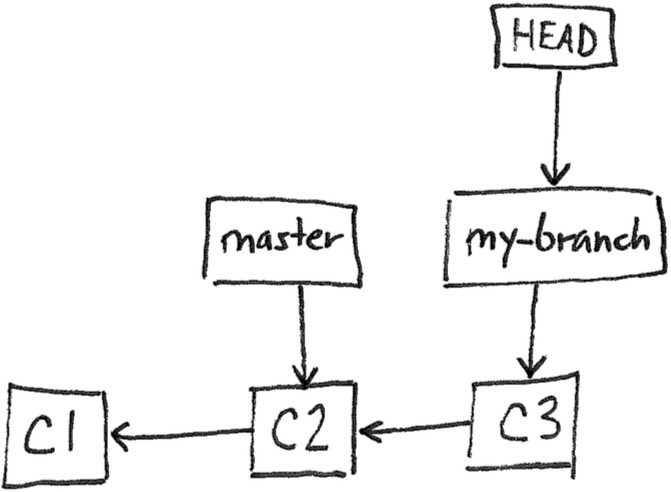
master

1. Imagine a Git repository (“repo”) with one commit and one branch (master). A user makes a new commit to the repo. Draw a box-and-line diagram of the repo like the diagrams shown in class. In- clude all commit nodes (with made-up hashes), all HEAD nodes, and all branch nodes.
2. Continuing the previous scenario, the user creates and checks out a new branch “my-branch” and then makes a new commit. Update the previous diagram to reflect these actions.Solutions:

1.



2.



# Multiple-Choice Questions:

Consider these four versions of a Rails model file, *cat.rb*:

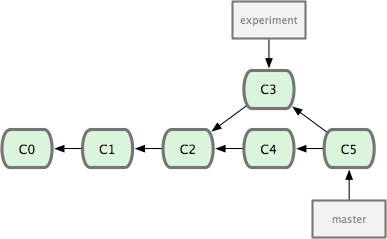
|  |  |
| --- | --- |
| **Version A** | **Version B** |
| **Version C** | **Version D** |

Now, imagine the following scenario. Alice and Bob are using GitHub to collaborate on a Rails project. They both are working in parallel on Version A of *cat.rb*. Each modifies the file *cat.rb* as follows: Alice changes it to be Version B, whereas Bob changes it to be Version C (note the difference in length maxi- mum).

Bob commits his changes first, and pushes them to the GitHub repo. Then, Alice commits her changes.

1. Which one of the following would happen if Alice next did a Git **push**?
   1. Her commit would be added to the GitHub repo, making the current version in GitHub be Version B
   2. Her commit would be merged with Bob’s in the GitHub repo, making the current version in GitHub be Version D
   3. Her push would be rejected, leaving the current version in GitHub as Version C
   4. Her push would be rejected, leaving GitHub unchanged, and Bob’s code would be download- ed and merged into her working directory, making her working copy Version D
   5. None of the above
2. Which one of the following would happen if Alice instead did a Git **pull**?
   1. Her pull would be rejected, leaving her working directory with Version B of *cat.rb*
   2. Her working directory would be updated to have Version C of *cat.rb*
   3. Her working directory would be updated to have Version D of *cat.rb*
   4. The GitHub repo would be updated to have Version B of *cat.rb*
   5. None of the above

Consider this Git repo.



1. Why does the C5 commit node point at both C3 and C4?
   1. Because C5 was created by merging data from C3 and C4
   2. Because C3 and C4 were each created by modifying data from C5
   3. This example is invalid: The node to which “experiment” directly points (C3) should not be reachable from the node to which “master” directly points (C5)
   4. This example is invalid: C5 should never point at both C3 and C4
   5. None of the above

# Solutions:

1. c
2. c
3. a

**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