**Git Commands**

1. Git clone <repository url> -> create a copy of the repository in our local machine
2. Git add filename.ext -> will add a new file to the repository, if you have made changes to the existing files of the repositories and do not want to include all the file names then just give the command as **git add .**
3. Git commit -m “message” -> files you want to commit to the repository
4. Git status -> will tell the status of the current repository
5. Git push -> push the changes of the repository
6. Git pull -> if changes are made by others in the remote repository, by using git pull you can pull down any of the latest changes
7. Git log -> list of all commits
8. Git reset --hard <commit> or git reset --hard origin/master
9. Git commit -am “message” -> by doing this, you can skip the git **add step**

**Merge Conflicts:**

If you make changes to a file of the repository in the local and someone made changes to the same file and same set of statements in remote. So if you try to git pull, then you will get a merge conflict error. And in the file you will be able to see the part of the code which you changed and the same part of the code which got changed by someone else in remote. So now, it is up to you to decide whose part of code you want to keep. So you can delete the other part. And then again do git add, git push, to push the code in the repository

**Reverting to previous commits:**

1. Git log will will list all the commits done previously with hashcode for each commit
2. Git reset --hard<hashcode\_Of\_a\_commit\_you\_want\_to\_go\_back\_to> 🡪 will revert back to the existing changes in local
3. **Git push --force**, will revert -> will revert back to existing changes in web.

**Branching:**

1. **Git branch** -> will show all the branches of your repository
2. **Git branch feature** -> will create a new branch named **feature**
3. **Git checkout feature** -> will switch to branch feature from master

Now new features, changes will be reflected in the feature branch, so now if you again switch back to master by giving the command – “git checkout master”, the new changes won’t be seen in the file. So if everything is ok and if you want to merge the feature branch to the master branch then the following command has to be given:

**Git merge feature**

1. **Now, if you want to push the branch to web from local machine, below commands should be used:**

**Git checkout features**

**Make changes to the files, and then the below commands to be used**

Git commit -am “Message”

**Git push –set-upstream origin feature**

**Git pull**

**In remote, there might be a lot of changes to application code, and in your local you might not have all the recent changes. So you write the below commands to fetch all the new changes and to merge those changes:**

**Git fetch**

**Git merge origin/master**

**So “git pull” is the shortcut for the above 2 commands**

**Fork and Pull Request**

In order to work on open source projects, you have to first fork the main code from the remote repository where the original project is present to your own repository. Then from your own repository use git clone “url” to clone it the code to your local machine. Create a branch say New\_feature using the following command :

git branch New feature

git checkout New\_feature

Make changes and the again push it to your own repository by

Git push origin/New feature or **Git push –set-upstream origin feature**

Now in Github navigate to the forked repository and click on **Pull Request** button

In Pull Request, mention the source branch as own New\_feature branch, the destination repository is the original guy’s repository and the destination branch is master.

Provide title and description for the pull request. You can put other people’s name in the reviewer section name who needs to approve the code.

After the Pull Request is done, then the original creator guy will click on Pull requests tab, if the code seems fine then by clicking on Merge button will approve the pull request and will merge the new feature into the master’s branch.

If some error needs to be fixed in the new\_feature branch then it can be done and push the changes into my own repository and the new commit will be automatically added to the original pull request.