**Git procedure (Distributed Version Control System)**

Local Changes -> To be added to staging -> To be committed -> To push to the remote repository.

**Basic Commands**

1. Git status -> To check the status of files.
2. Git config -> to check the basic configuration like email, username etc.
3. Git log -> to check all the commits and history.
4. Git add -A -> To add all the files into the staging area.
5. Git add .filename -> To add this particular file into the staging stage.
6. Git commit -m “message” -> To make a commit of the staged files

Before pushing any changes we should pull changes made in the remote repository.

1. Git pull origin master(default branch)
2. Git push origin master
3. Git branch -a -> To check the list of the branch.
4. Git branch branch\_name -> to create a branch.
5. Git checkout branch\_name -> to switch to the other branch.
6. Git branch --merged -> to check if branches are merged.
7. Git merge branch\_name -> to merge branch

**Rollback and mistakes commands**

1. Git checkout filename -> To rollback changes made in the local system in files.
2. Git commit amend -m “message” -> to change the message if the wrong message is typed in the last commit. It will change the message of the last commit w/o making a new commit. (Before pushing the changes in the remote repository.)
3. Git commits --amend -> If you introduced some changes/some new files and want to commit the changes in the last commit itself, then it will not create any new commit. It will make it as a part of the last commit itself.
4. Git log --stat -> It will give you the full information which files you committed.
5. Git cherry-pick #value\_of\_the\_commit -> If you made changes in another branch, want to make those changes in another branch. But it will not remove those changes from another branch we have to it manually.
6. Git reset #value -> It will roll back down to the mentioned id. (soft, hard default different reset options)
7. Git reflog -> It will show all the commits when it was last referenced. With all cherry-pick etc.
8. Git revert ->

**Stash Command - Stash command is useful when you don’t want to push your changes and want to keep them for future if want to make those changes otherwise you can flush them out.**

1. Git stash save “your message” -> It will save your current work.
2. Git stash list -> Will give you the stashed listed changes
3. Git stash apply stash{id} -> It will apply those changes in the file if you want to commit, but one disadvantage is that it will not remove those changes from stash.
4. Git stash pop -> It will apply the changes in the file if you want to commit and also remove the changes from stash.
5. Git stash clear -> It will remove all the stashes if you don't want to apply.