**Git and GitHub**

**Initializing git in local machine:**

1. Create a folder
2. Move into that folder
3. Initialize git in here using “**git init**”
4. Config the git using
   1. **git config --global user.name “Name”**
   2. **git config --global user.email “email address”**
5. “**git status**” command will give the file names that are updated or made any changes in that files. Or files that are present in **staging area**.
6. “**git add filename**” command will move all the updated or new files to staging area.
7. After moving all the files to staging area use command called “**git commit**”, it can be used in two different ways:
   1. **git commit -m “commit message”.** This command can be used for small commit messages.
   2. **git commit** After this default editor will open which will ask us to write the commit message. We will use this for writing detailed commit messages.
8. “**git log**” command will show the commit history of the repository.

**Staging Area:** after using command **git add** updated files or new files will move to staging area. After **git commit** command updated files or new files will move from staging area to repository. (one refer for more clarity).

**Head:** git uses **head** as a book mark, head represents the most recent or current snapshot (commit).

**git log -p** will give the detailed history of each commit. i.e. changes that are added and deleted in each commit will be showed using **git log -p** command.

**git show commitID** command will show detailed commit details of the given commit ID.

**git log --stat** will show the commit history with detailed numeric values, like no.of lines added and no.of lines deleted.

**git diff** command will show the changes made in the files before adding those changes to staging area (like confirming before staging the changes).

After adding the changes to staging area we can also check and confirm changes by using the command **git diff --staged**.

**git add -p** command helps us to reconfirm the changes made before staging the changes.

**git rm filename** command was used to delete files from repository.

**git mv oldfilename newfilename** command was used to rename the files or move data from old file to new file

**git checkout filename** command helps us to undo the changes made in the file. It only works before the file gets staged. (to do this the file must tracked one before itself).

**git reset HEAD filename** command helps to undo the changes even if the file get staged.

**git commit --amend** command helps us to modify and add changes to most recent commit. (Avoid amending commits that have already been made public).

You committed a change and you find that there was a bug in changes made in the latest commit. Consider v1 = original, v2 = original + changes, and committed v2. now you find that there was a bug in v2, now you want v2 to be rollbacked. **git revert HEAD** command will rollback the latest commit(v2) to v1 and it will commit automatically v1 again by itself, i.e., changes made in the latest commit will deleted and it will roll back to previous commit.

Now commit history will look like below:

Committed v1 (HEAD)

Committed v2

Committed v1