What is Git?

It is a tool which captures the local changes and keep it locally. Simply it is a Version Control System.

What is GitHub?

It is basically a Platform which provides us to put our codes there. It open chances of Collaboration with other companies and make projects together. GitHub also provides many other benefits.

Git Configuration Set-Up :-

- git config --global user.name (Your Name)
- git config --global user.email (Your Email)
- To check if your name is registered or not, enter the command- git config user.name
- To check if your email is registered or not, enter the command- git config user.email

Various Commands:-

- **mkdir:**make a directory/folder.
- **dir:** show that in which directory we are currently.
- **cd (file or folder name):** to go to other folder or file.
- **cd .. –** to go one file back.
- **ls:** show the list of all the files
- **ls –lart:** to see the list of hidden files.
- **git log:** shows the commit history.
- **git status:** shows the list that which files are staged or non-staged.
- **git init:** initializing the folder, and makes a repository (.git)
- **git add ./ git add -A :** to add all the folders or file.
- **git commit –a "message"/ git commit –am "message":** it is a checkpoint means it is used to save the changes locally.
- **touch (File Name):** to create a new file.
- **git checkout –b "new branch name":** to make a new branch.

- **git checkout –d “branch name”:** to delete any branch.
- **git checkout “branch name”:** to go to another branch.
- **git branch:** shows all the branches that are present.
- **git merge (branch name):** to merge two branches.
- **code . :** open VS Code on current folder.
- **code (File name.txt):** open VS Code on the file mentioned.
- **git reset:** deletes all the log permanently.
- **git reset --hard (commit Id):** it deletes a particular commit permanently.
- **git revert (commit Id):** creates a new commit that effectively cancels out the change made in target commit.
- **.gitignore:** to create a file which is not tracked by git.
- **git clone (url):** to bring any repository from github to your system. [url : uniform resource locator]

- **git remote –v:** shows the list of all the urls that is available to push or pull [-v stands for verbose].
- **origin:** it is url of a repository in github.
- **git diff:** tells the difference between the branches.
- **git pull:** update the changes in the repository.
- **git push:** it sends the updated repository to our github account.
- **git push origin (branch name):**

How to merge two branches?

1. In default, we are in Master Branch.
2. Now we have to make another branch by “git checkout –b (branch name)”.The new branch will be created.
3. Then we will check branch status that in which branch we are currently by using the command “git branch”.
4. Let say if we are in Master branch right now, then open VS Code for it using “code

(filename.txt)".Enter something in it then save it and close the VS Code.

5. Now commit the new thing we added.
6. After that, change your branch and go to other branch by "git checkout (other branch name)".
7. Now again we will open the VS Code with same file and write some different text this time.Save it.
8. Commit the changes .
9. Now we have to merge both the branches so for that we will enter "git merge (branch name except current branch)".

How to make Pull Request?

1. Make a new folder at desktop and add it in git.
2. Then we will go to that repository in which we have to make the changes and bring it in our repository by doing "fork".

3. Then we will go to our github and copy the url of it.
4. Then go to git and open the VS Code and new folder which we made.Now to locally bring the repository folder ,we will use the command “git clone (copied url)”.
5. Now after successfully cloning,enter inside the file in which we have to make changes by “cd (file name)”
6. Now “git pull” to update any changes in file (if any).
7. Now make a new branch and enter that branch.
8. Now do whatever changes you want to do in the file and save it (cltr s).
9. Add it using git add –A.
10. Now we will commit the changes that are done –“git commit –am (message)”.
11. Now our work is done .Check your origin by “git remote –v”.

12. Then we will push the changes back to our github account by entering the command “git push origin (branch name)”.
13. Then go to github account and click on ‘compare pull request’. If you want to add the description then add else click on ‘Create pull request’ Button.

- **Report compiled by Shiwangi.**