GIT

1. Git init –> to create empty local repository
2. Git clone –> to clone project from remote repository to local repository
3. git config --global user.email "email id" -> email id for commit record
4. git config –global user.name “name” -> user name for commit record

Basically, there are 3 places to store these variables:

* System -> These variables are available for every user in the system and stored in C:/Program Files/Git/etc/gitconfig
* Global -> Global configurations are available for the current user for all the projects and stored in C:/Users/Username/.gitconfig
* Local ->  Local configs are available for the current repository only and stored in C:/Users/MyProject/.git/config

1. Git add <file name> –> to add specific file from working directory into staging area
2. Git add . –> to add all files from working directory into staging area
3. Git add \*.pdf -> we can use pattern to add files
4. Git restore - -staged . -> to unstage the added files
5. Git restore - -staged <file name> -> to unstage the added file
6. Git commit -m”message” –> to add staging area files into local repository
7. Git log -> to check all the actions in commit stage
8. Git log <file name > -> to check specific changes
9. Git log - - oneline -> to get concise message
10. Git log -n <number> -> to get n number of first commits
11. Git log -n <number> --oneline -> get get n number of first commits by one line
12. Git log - - grep “search message” -> we can search commit message by search message
13. Git commit -a -m “files modified” -> this command is only useful in modified files, (this is the combination of git add + git commit) we can even use – git commit -am “ “
14. Git status -> this command gives detailed information about file status
15. Git status -s -> this command gives concise (single line) message
16. Git config –list -> to get all configurations
17. Git diff <file name> -> to check difference between working directory changes and staging area file or local repository
18. Git rm –staged <file name> -> to remove file from staging area
19. Git checkout <file name> -> to discard changes in working directory copy
20. git checkout -> To discard changes in all tracked files of working directory.
21. Git reset -> to remove files from staging area and move back to working directory
22. Git reset - -mixed-> changes will be discarded in local repo and staging area. It won't touch working directoy. Working tree won't be clean. But we can revert with git add . git commit
23. Git reset - -soft-> Changes will be discarded only in local repository. It won't touch staging area and working directory. Working tree won't be clean. But we can revert with git commit
24. Git reset - -hard -> Changes will be discarded everywhere. Working tree won't be clean