1. Git install (search)
2. Click on Git SCM
3. git
4. git --help
5. git help -a
6. git help -g
7. git help <command>g
8. git config --global user.name "bilal khan" (to give user name and email)
9. git config --global user.email [er.bilalkhan007@gmail.com](mailto:er.bilalkhan007@gmail.com)
10. git config –-global alias.st status (to use alias)
11. git config –list or

git config user.name

git config user.email (to check user name and email)

1. Three stage architecture
   1. Working directory
   2. Staging area
   3. Repository (git directory)
2. git status
3. git init (make folder git repository)
4. rm -rf .git (to delete repository)
5. pwd (present working directory)
6. ls (list content of directory)
7. git add –-a (send all files in stage area)

Or

Git add .

1. git restore –-staged file name (remove all files from stage area)
2. git commit -m ‘first commit’
3. ( change the file )
4. git status
5. git add ‘name of file’ (to update what to commit)
6. git add –-a (send all files in stage area)
7. git commit -m ‘second commit’

Or

Git commit -a -m ‘second commit’

1. git log (details of all commits)
2. git log -p (to see all commits with changes)
3. git log -p -2 (to see 2 commits with changes)
4. git log –stat (to see commits summary)
5. git log --since=2.days (to see commits of last 2 days)
6. git show “name of commit”
7. Touch .file name (to create untracked file in pwd)
8. Touch .gitignore (creates untraked file)
9. Put names of untracked file in .gitignore to ignore them
10. git rm .gitignore (removes gitignor file from repo)
11. Press q to quit : (to quit from pager)
12. Exit (to close gitbash)
13. Go to github and copy repository link
14. git clone ‘link’ (to pull remote repository)
15. After files added in staging area, if we modify the file then
    1. Commit will ignore the changes
    2. To save changes first add again in staging area and then commit
16. git diff (compares working directory with staging area)
17. git diff -–staged (compares working staging area with latest commit)
18. git mv oldfilename newfilename (rename file and stage)
19. git rm --cached file name (untrack file)
20. git restore file name (to undo changes in modifiedfile as per last commit)
21. git checkout – file name (to undo changes in file as per last commit)
22. git checkout – f (to undo changes in all files as per last commit)
23. git remote
24. git remote add origin URL (giving name to URL)
25. git remote remove origin (deleting origin)
26. git push -u origin master (to push changes in origin master on github )
27. git checkout -b name of branch
28. git branch
29. git checkout master
30. git merge name of branch (to merge branch. You should be in master branch)
31. git branch -–merged (gives merged branches when viewed from main)
32. git -–no-merged (gives unmerged branches when viewed from main)
33. git branch -v (gives last commit of branches)
34. git branch -D name of branch (deleting unmerged branch)
35. git branch -d name of branch (deleting merged branch)
36. git push origin branch name (to add branch on github)