(3,

1)Check version:

git –version

2)Configure:

Global: git config --global user.name "Zahid Hasan Jamil"

git config --global user.email "mzhj19@gmail.com"

Local: git config user.name "Zahid Hasan Jamil" // local means one folder or like this

git config user.email "mzhj19@gmail.com"

3)Show configuration:

git config –list

4)Change configuration:

git config –global user.name “new name” // same for others

cd .. // previous folder

mkdir // make directory/folder

touch day1.txt // make file

open day1.txt // open file

5)Initialize:

git init

6)Check status // staged or unstaged, tracked or untracked

git status

\*\*\*\* ls // only visible file

Ls -a // all files ,with hidden

Version control possible from local repository ,Not from working

Directory or stagging area

Mkdir // make directory

rm -rf test.txt // remove/ delete a folder ////// rm =remove directory,rf=remove file

\*\*\*\*

7)Working directory to stagging area:

git add day1.txt // single file

git add . // all files in directory but not subdirectories

git add -A // A for all, all of directory and subdirectories

git add \*.js // all files of .js extention ,not subdirectories

git add \*\*/\*.js // all of directory and subdirectories

8)Unstaged from staged:

git rm –cached day2.txt // rm for remove

9)Check difference of previous staged and current unstaged file:

git diff

10)Restore previous:

git restore day1.txt

11)Commit: Moving staging to local repository

git commit -m “Commit massage”

\*\*\*\*\*\*\*\*

Stagging and commit at a same time directly:

git commit -am “message”

Or, git add . && git commit -m “message”

\*\*\*\*\*\*\*\*

12)Uncommit: Local repository to staging area

git reset --soft HEAD^ // local repository to staging area , HEAD means latest commit

git reset HEAD^ // local repository to working directory

git reset –hard HEAD^ // reseting everything to previous commit

\*\*\* git reset –soft HEAD-2 // HEAD~2 // first 2 head commit

13)Undo/discard: Staging area to working directory

git checkout day2.txt

git checkout 3edde3f // id of going to which commit, id’s after that the commits will be discard

git checkout master // to recent commit

13) Show the commit:

git log //details

git log --oneline // in one line

git show 2e232da // id from online command

14).gitignore:

.env // .env

\*.txt

!story.txt

Folder\_name/ // folder ignore

15)Check remote connection:

git remote -v // shows the remote along with the url

16)Connect with remote repository/ Local repository to remote repository connection:

git remote add origin <https://github.com/mzhj19/gitGithub.git>

/// origin =any name=the url,, like c++ define

17)Clone git repository:

git clone <https://github.com/mzhj19/gitGithub.git>

\*\*\*\*\*\*\*\*\*\*\*\*

If push into main not working ,push into master working:

git pull origin main –allow-unrelated-histories

git push origin main –force

Run git push -u origin master **instead of** git push -u origin main

Or **Solution B** - if you want to name the branch main

Run git checkout -B main **before** git push -u origin main

\*\*\*\*\*\*\*\*\*\*\*\*

18) Push/local to remote repository:

git push -u origin main

19)Branch:

git branch myBranch // create branch

git checkout myBranch // move to myBranch branch

git chechout master // move to master branch

git merge myBranch // merge with master

git branch -d myBranch // delete branch

20)Show branch:

git branch

27/11