To find help

git –help

or go to <https://git-scm.com/docs/git-config>

make directory

mkdir git

go into the directory in cmd

cd git

go to directory in git bash

cd /c/users/alanc/git

initialising an empty folder /repos as a SCM

git init

<https://www.youtube.com/watch?v=8JJ101D3knE>

can install git bash for a nice interface of git

lets say we create a file and write some content into the file

echo hello > file1.txt

echo hello > file2.txt

to edit in notepad

start notepad index.html

to track/check status of the file

git status

then we will realise all files not added to branch

To let GIT track them

Use add command

Git add file1.txt

Git add file2.txt

Can also type add file1.txt file2.txt

To put more words or append into the file1.txt

Echo world >>file1.txt

Check status

Git status

If someone check and think the add will cause more trouble to the code and all agree to revert

Git Restore file1.txt

To permanently store in git repository

Git commit –m “….your message..”

Once done they will tell you how many lines inserted and files committed

Make some change like type “bug fixer” in file 1

Then type git add file1.txt

If file2 is not needed as the system is removed and we don’t need the code in the **repository**

Git rm file2.txt

For files not added via the git add command but manually, it cannot be remove by above command until git add file3.txt is done and git commit

To check for files in

git ls-files

to rename files with cmd, it go into staging. then you need to add into repository after it is in staging

rename file1.txt main.js

then we want to stage it into the staging environ with git add file1.txt. git add main.js

to directly rename file to the git repository, assuming all team members agree, you can type in cmd

git mv main.js to file1.js

git

git status

ignore files

mkdir .gitignore

Git commit –m “add gitignore”

Remove folder accidently in staging

git rm --cached -r log/

to view previous committed log

git log

to really go to the previous state

git reset “xxx the number in the git log”

git reset “385e6e9c3fd7953d2fe8501f05f7d8a1294773e9”

To create a sub branch from the master branch to do a subtract function and multiply

git branch subtract

git branch multiply

to see all branches type

git branch

IF WE SEE THE \*AND GREEN, IT MEANS MASTER IS CHECKED OUT

just to see and review OTHER BRANCH

git checkout subtract

to see content of a file in git bash

cat index.html

to edit in notepad

start notepad index.html

to merge branch forward merge approach

switch to master branch then merge with the sub branch

git merge multiply

“write ur merge message

Press esc

Type :wq at the end

To delete sub branch that we have completely merged with main branch

Git branch –d multiply

To establish connection with github from git then push master to git hub

git remote add origin https://github.com/alanchow85/newrepo.git

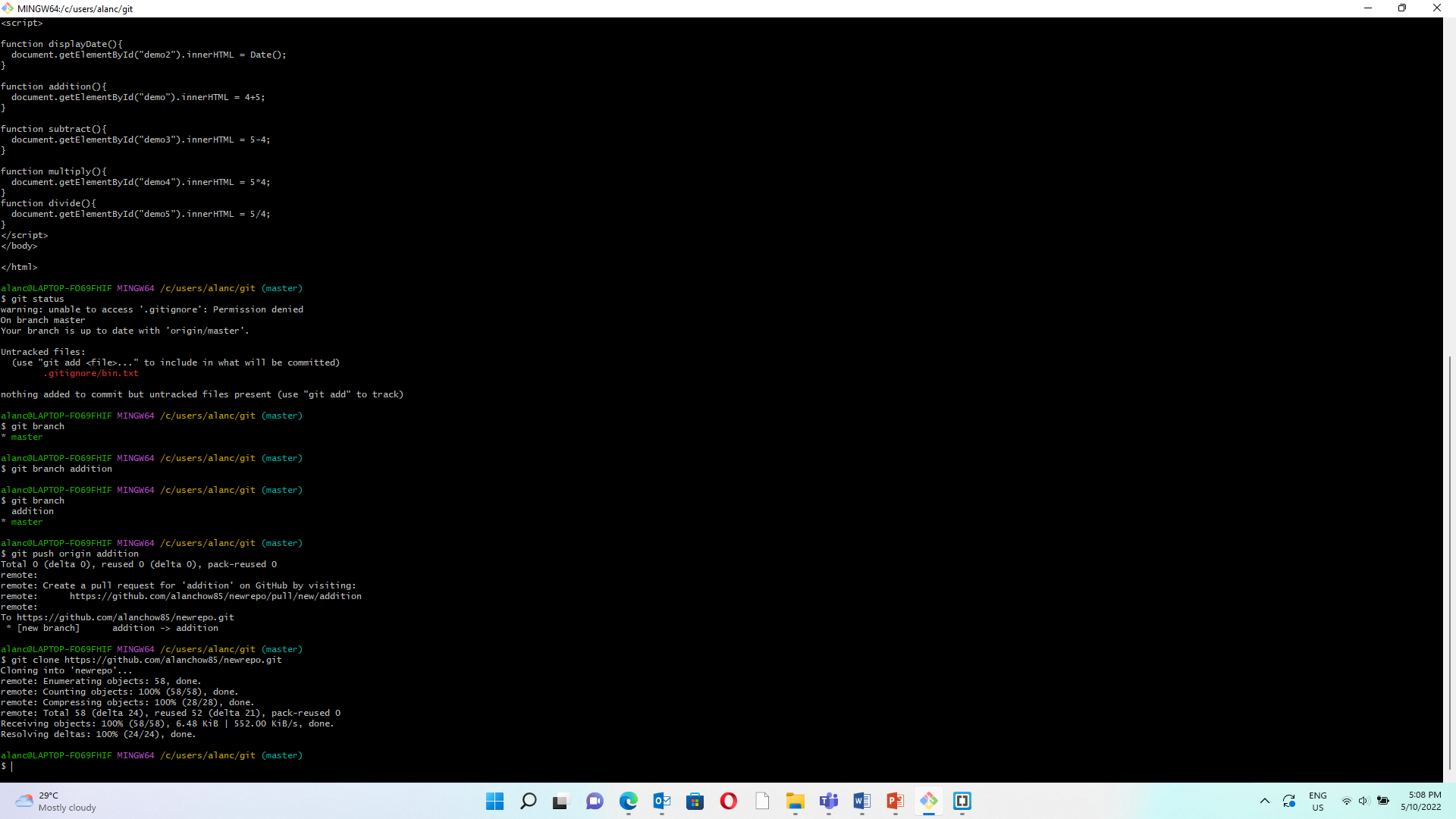
git push -u origin master

If conflict use force push

git push --force origin master

To clone github things

git clone https://github.com/alanchow85/newrepo.git



to restore to previous commit

git revert hashcode

