**Aaron Lim**

**005893468**

**CSE408**

**Lab#3: Git Hub**

**CSE 408- Lab #3**

**Version Control System**

**GitHub**

**Due: Thursday April 30th 2020, 8:00 am**

**Part 1:**

**(25 pts) Create a Github account (if you don’t already have one) and use the terminal to create a repository and upload a file.**

1. After creating a github account, go under the “+” and select “new repository”.

-Created new repository on github.com and named it “alim1091/lab3”

2. Open your terminal

3. Create a folder where you want to store your git repository work and place any file(s) you want in it.

-created folder on desktop and use git bash to navigate to the directory



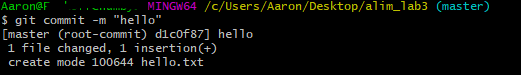
4. In the command line of the folder run “git init” to initialize it as a git repository.

-Accidently left out of terminal image, but the folder was initialized with a git

5. Next in the command line run either “git add .” or “git add your\_file\_name”. Using the period after add includes all the files in the repository/current folder.



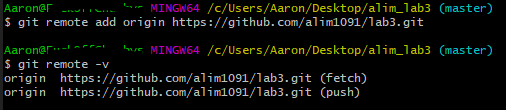
6. Afterwards run the command “git commit -m “whatever message you want to write to appear with the upload””



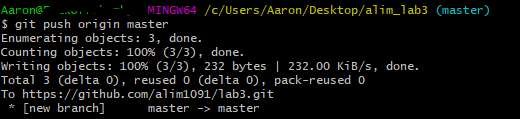
7. Go back to the repository on github.com you made and copy the link it has provided since it’s still currently empty.



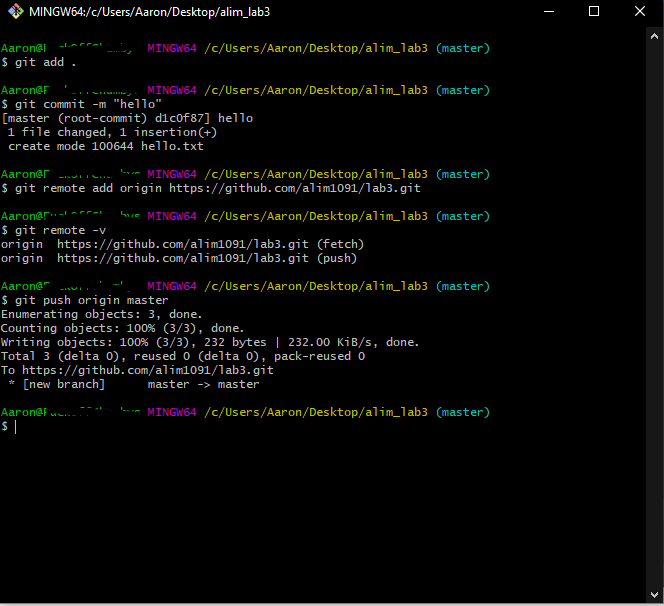
8. Run “git remote add origin url\_your\_copied”



9. Finally run “git push -u origin master” and you should see your file(s) uploaded to your repository.



**CMD:**



**Part 2: (30 pts)After uploading a file, create a new branch and then modify the file and merge it with the master branch**

1. In the command line now run the command “git checkout -b your\_new\_branch\_name”. This command switches and creates a new branch at the same time.

 named “lab3\_2”

1. Use the command “git branch” to confirm your in the new branch.



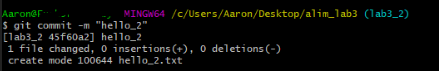
1. Add new random files to folder containing your repository.

-“hello\_2”

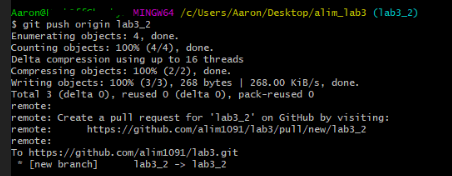
4. Run “git add .” or “git add your\_file\_name”.



1. Followed by “git commit -m “whatever message your want to appear with your new upload”



1. Then run “git push origin your\_new\_branch”.



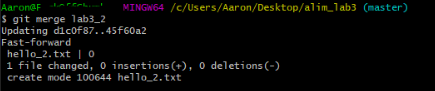
1. Now use “git checkout master” to swap into the master branch.



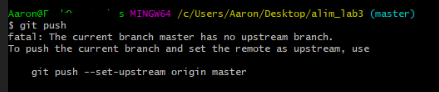
1. Verify by running “git branch”



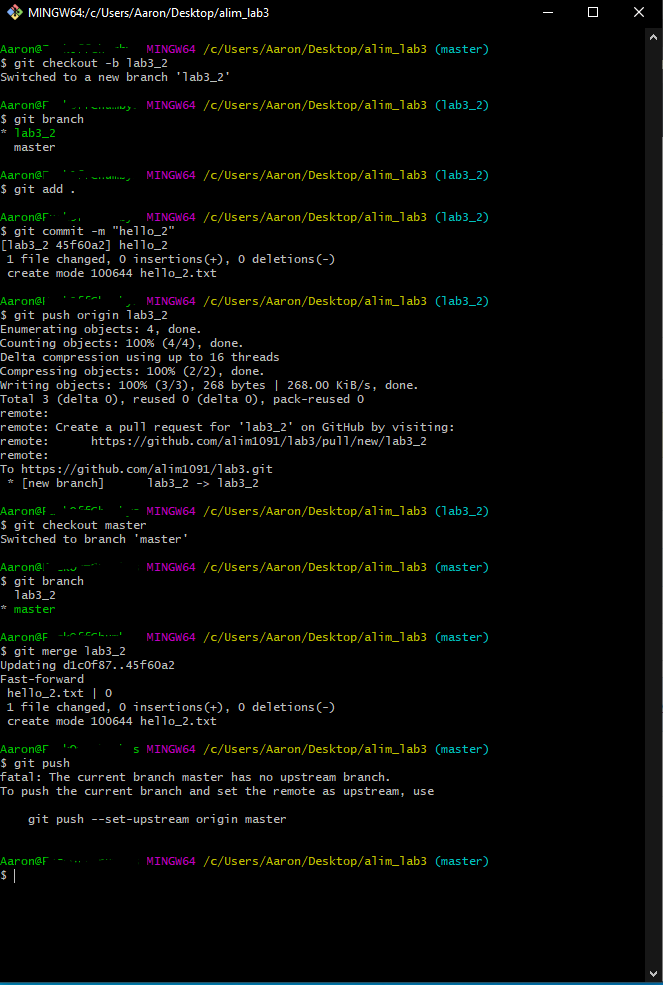
1. Afterwards run “git merge your\_new\_branch”



1. If no errors, run “git push” and you should see your items from your new\_branch appear in the master branch.



**Part 2 CMD:**



**Part 3: (25 pts)Clone a repo from a different github user (https://github.com/Juan-Inzunza) and perform a pull request.**

1. Create a new folder on your computer to hold the cloned repository.

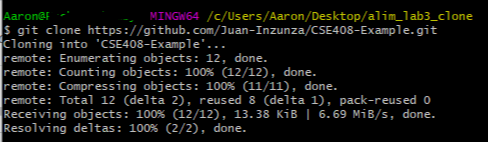
-Created folder “alim\_lab3\_clone”

2. Next open up the terminal and head to that new folder

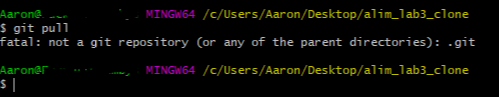
3. Go to https://github.com/Juan-Inzunza and select any of the repositories.

4. Once in the repo, hit the green “Clone or Download” button, and copy the link in the pop-up box.

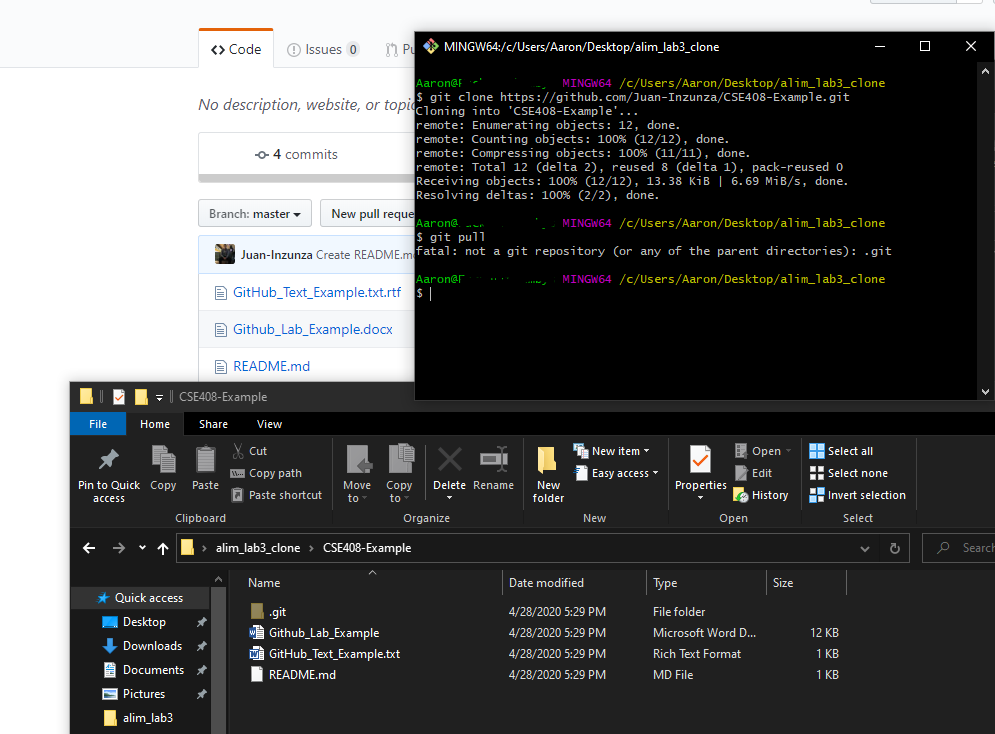
5. In the terminal run “git clone copied\_repo\_url”



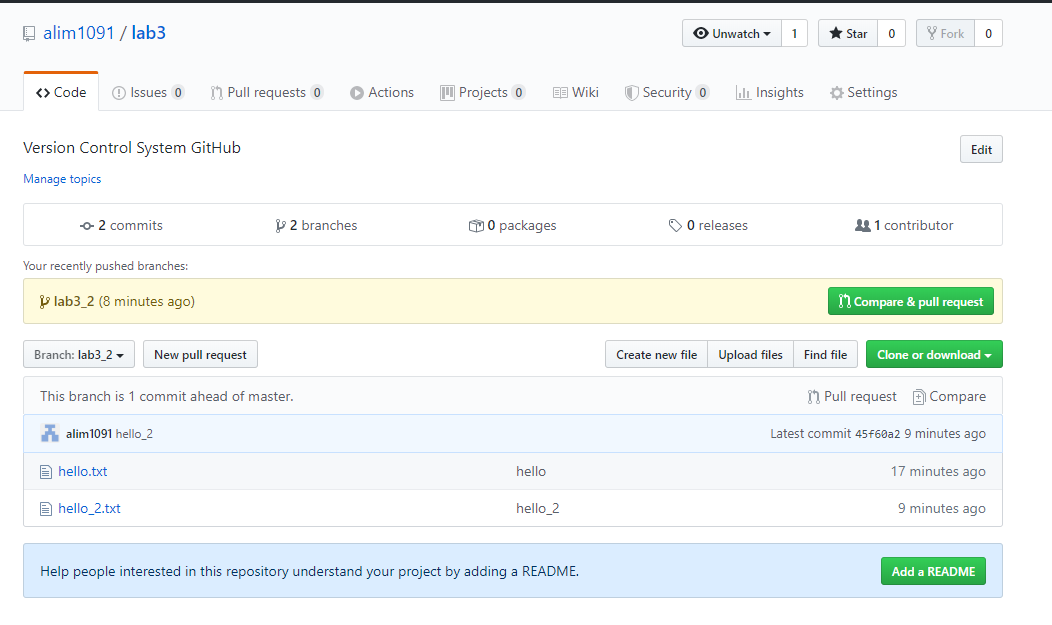
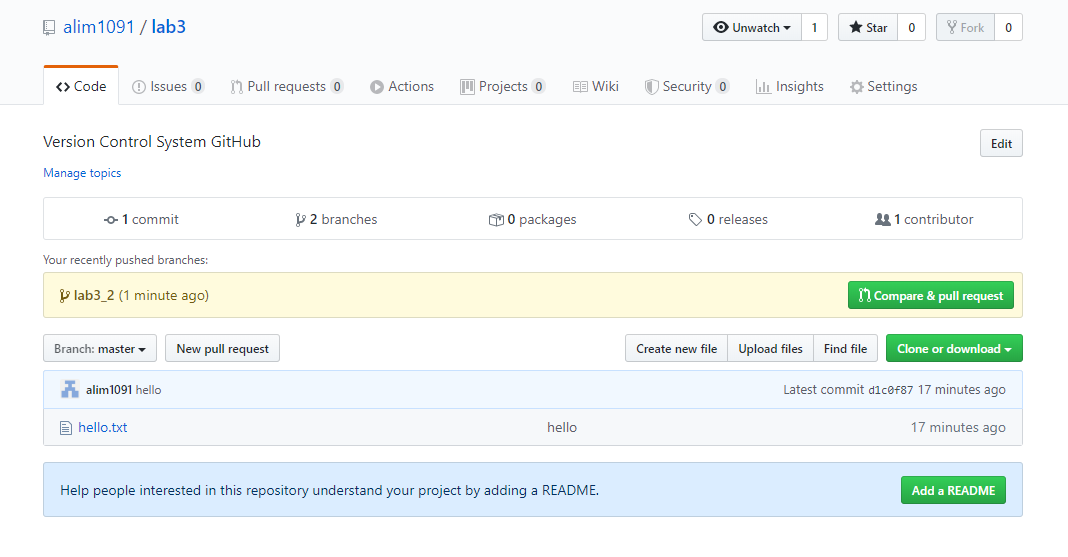
6. Then run “git pull” and you should see a copy of the repo contents in your folder



**Part 3 CMD:**



**Repositories:**



**Answer the following questions:**

(5) What benefits would a large team of developers get from version control? Identify at least two.

-compare versions of files with each other

-merge any file changes you make as a team

(5) What benefits would a single developer (working alone) get from version control? Identify at least two.

-Able to view different version of code he/she created.

-Change, merge, or recall files later

(5) What kind of files should you put in version control?

-Source Code

- Scripts

-Documents (word, pdfs, text)

-Tool Config files

(5) What kind of files should you not put in version control? Why?

-IDE config files, generated files, pre-compiled source code, binary documents/content