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

```
Aaron@I  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
```

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.

```
Aaron@I  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git add .
```

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

```
Aaron@F  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git commit -m "hello"
[master (root-commit) d1c0f87] hello
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

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

```
https://github.com/alim1091/lab3.git
```

8. Run "git remote add origin url_your_copied"

```
Aaron@F  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git remote add origin https://github.com/alim1091/lab3.git

Aaron@F  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git remote -v
origin  https://github.com/alim1091/lab3.git (fetch)
origin  https://github.com/alim1091/lab3.git (push)
```

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

```
Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/alim1091/lab3.git
 * [new branch]      master -> master
```

CMD:

```
MINGW64:/c/Users/Aaron/Desktop/alim_lab3

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git add .

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git commit -m "hello"
[master (root-commit) d1c0f87] hello
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git remote add origin https://github.com/alim1091/lab3.git

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git remote -v
origin  https://github.com/alim1091/lab3.git (fetch)
origin  https://github.com/alim1091/lab3.git (push)

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/alim1091/lab3.git
 * [new branch]      master -> master

Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$
```

<u>Part 2:</u> (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.

```
Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git checkout -b lab3_2
Switched to a new branch 'lab3_2'
```

named “lab3_2”

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

```
Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git branch
* lab3_2
  master
```

3. Add new random files to folder containing your repository.
-“hello_2”

4. Run “git add .” or “git add your_file_name”.

```
Aaron@[redacted]  MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git add .
```

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

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git commit -m "hello_2"
[lab3_2 45f60a2] hello_2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello_2.txt
```

5. Then run “git push origin your_new_branch”.

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git push origin lab3_2
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 268 bytes | 268.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'lab3_2' on GitHub by visiting:
remote:      https://github.com/alim1091/lab3/pull/new/lab3_2
remote:
To https://github.com/alim1091/lab3.git
 * [new branch]      lab3_2 -> lab3_2
```

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

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git checkout master
Switched to branch 'master'
```

7. Verify by running “git branch”

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git branch
  lab3_2
* master
```

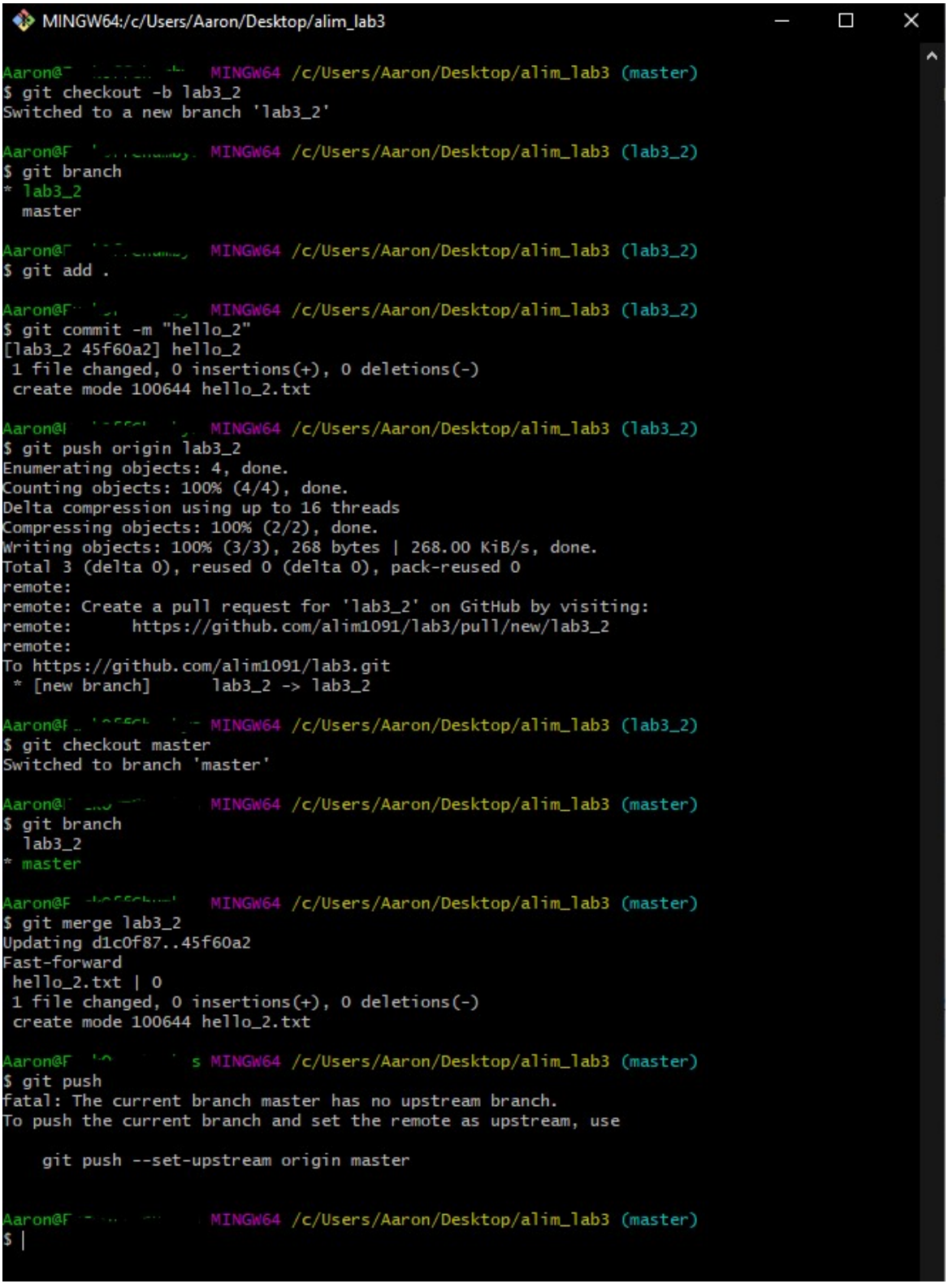
8. Afterwards run “git merge your_new_branch”

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git merge lab3_2
Updating d1c0f87..45f60a2
Fast-forward
 hello_2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello_2.txt
```

9. If no errors, run “git push” and you should see your items from your new_branch appear in the master branch.

```
Aaron@[redacted] MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
```

Part 2 CMD:


MINGW64:/c/Users/Aaron/Desktop/alim_lab3
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git checkout -b lab3_2
Switched to a new branch 'lab3_2'
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git branch
* lab3_2
master
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git add .
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git commit -m "hello_2"
[lab3_2 45f60a2] hello_2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello_2.txt
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git push origin lab3_2
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 268 bytes | 268.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'lab3_2' on GitHub by visiting:
remote: https://github.com/alim1091/lab3/pull/new/lab3_2
remote:
To https://github.com/alim1091/lab3.git
* [new branch] lab3_2 -> lab3_2
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (lab3_2)
$ git checkout master
Switched to branch 'master'
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git branch
lab3_2
* master
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git merge lab3_2
Updating d1c0f87..45f60a2
Fast-forward
hello_2.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello_2.txt
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use
git push --set-upstream origin master
MINGW64 /c/Users/Aaron/Desktop/alim_lab3 (master)
$

<u>Part 3:</u> (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”

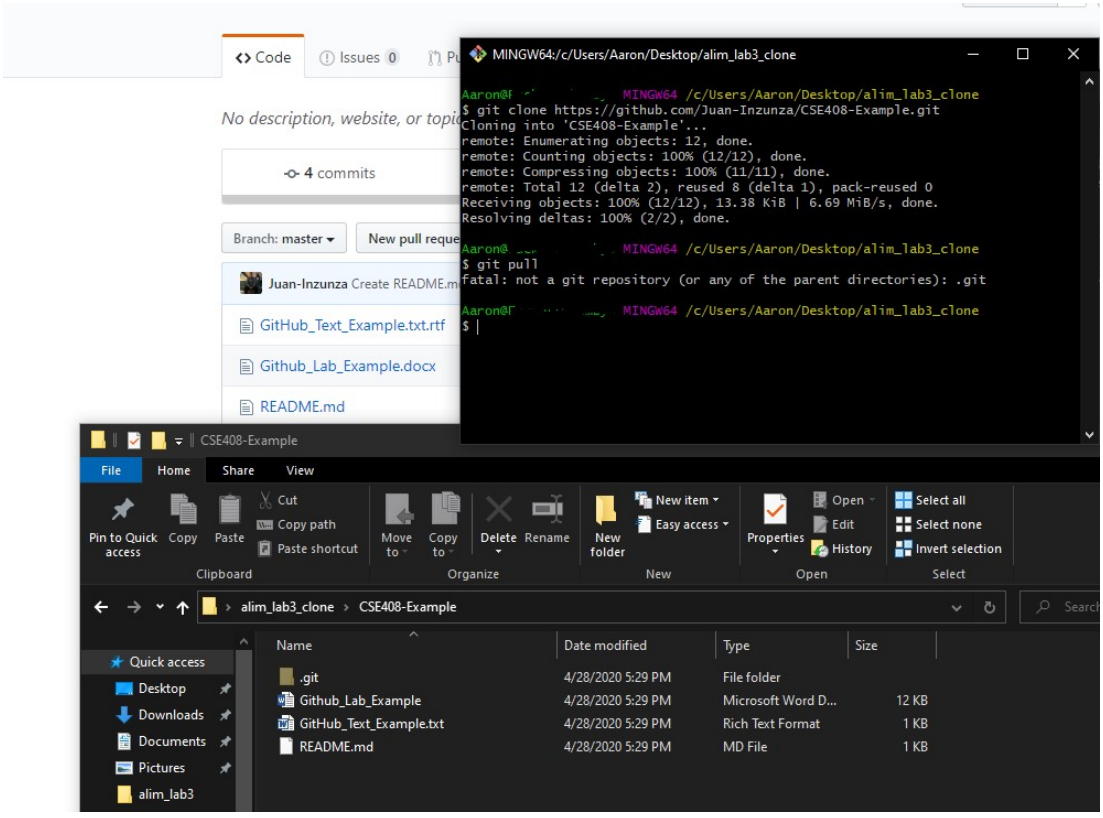
2. Next open up the terminal and head to that new folder
3. Go to https://github.com/Juan-Inzunza and select any of the repositories.
4. Once in the repo, hit the green “Clone or Download” button, and copy the link in the pop-up box.
5. In the terminal run “git clone copied_repo_url”

```
Aaron@ MINGW64 /c/Users/Aaron/Desktop/alim_lab3_clone
$ git clone https://github.com/Juan-Inzunza/CSE408-Example.git
Cloning into 'CSE408-Example'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 2), reused 8 (delta 1), pack-reused 0
Receiving objects: 100% (12/12), 13.38 KiB | 6.69 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

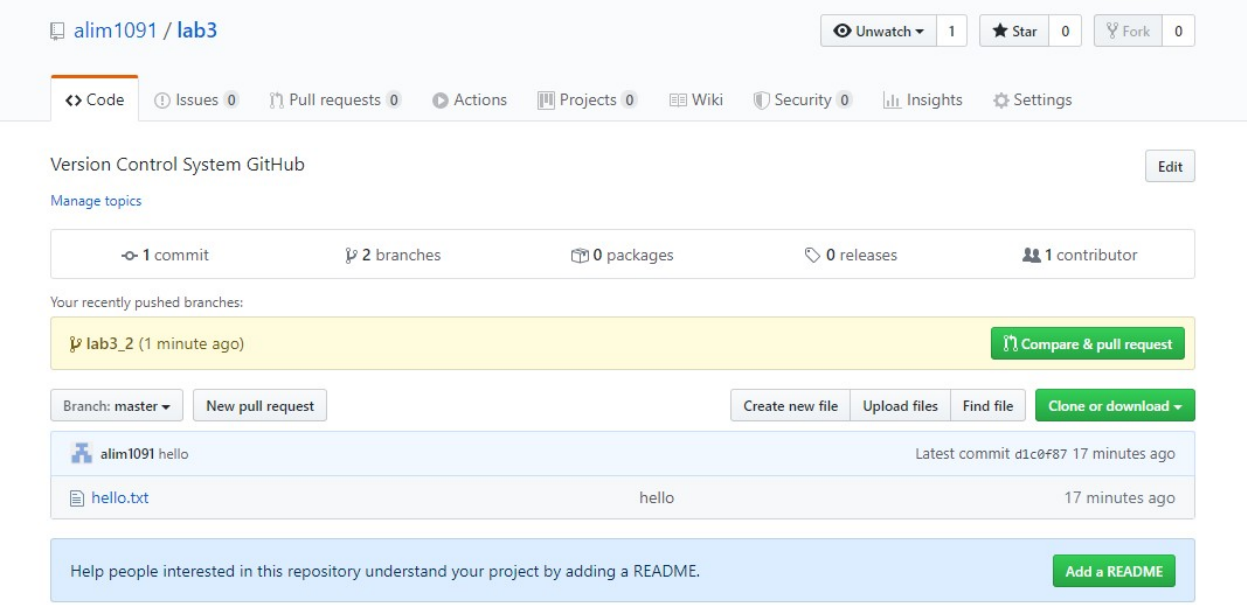
6. Then run “git pull” and you should see a copy of the repo contents in your folder

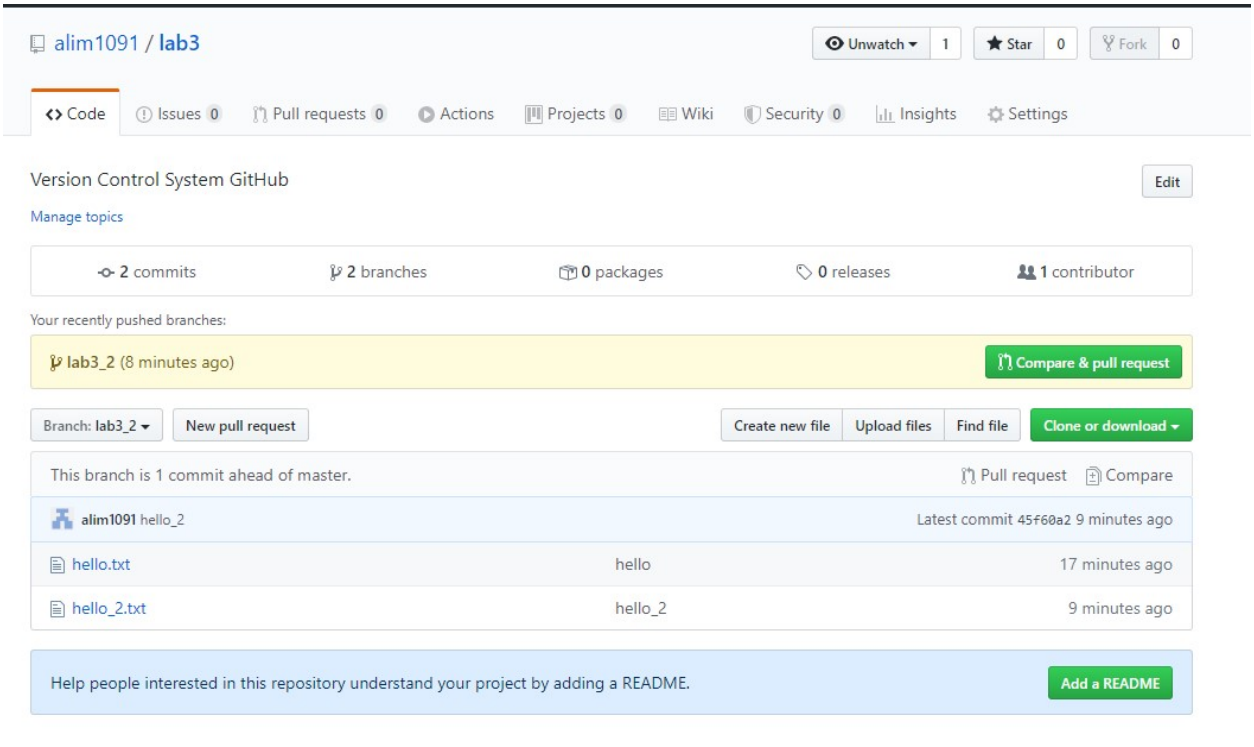
```
Aaron@ MINGW64 /c/Users/Aaron/Desktop/alim_lab3_clone
$ git pull
fatal: not a git repository (or any of the parent directories): .git

Aaron@ MINGW64 /c/Users/Aaron/Desktop/alim_lab3_clone
$
```

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