CS 458 - Homework 2

Deadlines

(Problem 1 should have been completed during the Week 2 Lab on 2017-08-30)

Problems 2 onward are due by 11:59 pm on Friday, September 8, 2017

Purpose

To help you to start getting to know at least some `git` command-line-level basics.

How to submit

(Problem 1 was completed and submitted during the Week 2 Lab on 2017-08-30.)

Submit your files for Problem 2 onward for this homework using `~st10/458submit` on **nrs-projects**, with a homework number of `2`

Instructions for using the tool ~st10/458submit

- If they are not already on nrs-projects, transfer your files to be submitted to a directory on nrs-projects.
  - (Use `sftp` (secure file transfer) or an `sftp` application such as WinSCP or FileZilla, connecting to host `nrs-projects.humboldt.edu`, and then transferring the desired files.)
- Once all of your files to be submitted are in a directory on nrs-projects, then use `ssh` (or Putty in a campus Windows lab) to connect to `nrs-projects.humboldt.edu` .
- use `cd` to change to the directory containing the files to be submitted -- for example,

  ```
  cd 458hw02
  ```

- type the command:

  ```
  ~st10/458submit
  ```

  ...and give the number of the homework being submitted when asked, and answer that, `y`, you do want to submit all of the files with of-interest-to-458 suffixes in the current directory. (It is fine if a few extraneous files get submitted as well -- I'd rather receive too many files that too few, and typing in all of the file names for each assignment is just too error-prone!)
- you are expected to carefully check the list of files that the tool believes have been submitted, and make sure all of the files you hoped to submit were indeed submitted! (The most common error is to try to run `~st10/458submit` while in a different directory than where your files are.)

Important notes

- FUN `git` FACTS that did not come up during the Week 2 Labs:
  - If you have created a repository by **cloning** it from another repository (using `git clone`), then you can get any updates from that original repository, and update your cloned copy accordingly, with the command:

    ```
    git pull
    ```

Problem 1

...was completing, as a team, the in-lab activity during the Week 2 Lab on Wednesday, August 30, creating the file `458lab2.txt` with contents as described during lab, and submitting ONE copy per team during lab with a homework/lab number of **82**.

Problem 2

In a file `458hw2-2.txt`, first put your name and the last modified date.

Then, to help encourage everyone to get "up to speed" on `git`, answer each of the following questions, preceding each by the problem part it is the answer for. (Note: along with this homework handout, there is also a "GitHub Git Cheat Sheet" handout; this may be useful along with the lab projected notes, the link on the main public course web site to Scott Chacon's "Pro Git" online book, and a variety of resources from github.com. Also see the **Important Notes** section above!)

2 part a

Assume that, at some point in the past, you have cloned your team's private repository on GitHub into a local repository on your own computer.

Give the command you can now type to get the newest updates from the team repository on GitHub, (changing your local copy).

2 part b

Assume that you have modified a file in your local repository named `Widget.cpp` so that it now implements a user story **Add widget colors**. Give the command you can now type to *stage* your modified `Widget.cpp` for a commit.

2 part c

Give the command you can now type to commit all of your repository's currently staged files, all happening to have to do with adding colors for widgets, including an attempt at an appropriate descriptive log message.

2 part d

Finally, give the command you can now type to push your committed changes to the team's private repository on GitHub.

2 part e

Give the command you can use at any time to check the current status of your repository, to see what, if anything, has been modified.

2 part f

Give the command you can use at any time to see a log of the commits to your repository.

Submit your resulting file `458hw2-2.txt`

Problem 3

(By the way -- if you have not yet accepted your GitHub invitation for the organization `hsu-cs458-f17`, I think you will have difficulty completing this problem! Please accept it IMMEDIATELY, if you have not already, by going to `https://github.com/hsu-cs458-f17`)

As you know, you have been (invited to be) added to organization `hsu-cs458-f17` at `github.com`. You have now also been (if you accepted that organization invitation) added to a team `cs458`. Team `cs458` has access to a repository `458hw02`.

`git` is installed on nrs-projects; to provide some `git` practice that does not involve your team's eventual repository, you are going to use repository `458hw02` from nrs-projects (and it is up to you where you use Git for your team project work, of course! The beauty of Git is that you can work locally on own computer, or even multiple computers, or nrs-projects, etc.)

(Note: some of the commands given here and in the next few problems will prompt you for your password. That should be your GitHub password.)

As recommended in our Git intro, and on the Git Cheat Sheet from GitHub, and in Section 1.5, pp. 10-11 in Scott Chacon's "Pro Git", you should set your username and e-mail address for your repositories. Practice doing this on nrs-projects if you did not already do this during lab:

```
git config --global user.name "yourFirst yourLast"

git config --global user.email yourEmail@humboldt.edu
```

IF you would like to specify a default text editor that will be used when Git wants you to type in a message, you may do so using (Pro Git, p. 11) -- it notes that, on many systems, the default editor is `vi` or `vim`.

```
git config --global core.editor desiredEditorName
```

To show that you've done the above on nrs-projects -- to see for yourself, AND to make a file to submit to me -- now do the following two commands:

```
git config --list
```

```
git config --list > 458hw2-3-config-list.txt
```

Submit your resulting file `458hw2-3-config-list.txt`

Problem 4

Now that you have configured those settings, clone the `458hw02` repository that GitHub team `cs458` has access to onto nrs-projects. Run this command on nrs-projects (from wherever you would like this repository to be within your account on nrs-projects):

```
git clone https://github.com/hsu-cs458-f17/458hw02.git
```

...and enter your GitHub username and password when prompted.

Demonstrate that you succeeded by running the following commands (STARTING in the nrs-projects directory in which you RAN the above command, which should be the one now CONTAINING your cloned repository directory) (note that >> appends to a file...)

```
echo yourName >> 458hw2-4-show-cloned.txt
```

```
pwd >> 458hw2-4-show-cloned.txt
```

```
ls 458hw02/* >> 458hw2-4-show-cloned.txt
```

Submit your resulting file `458hw2-4-show-cloned.txt`

Problem 5

Here is where we are going to live dangerously!

Now do:

```
cd 458hw02
```

...so you are in your cloned GitHub repo. You'll see there is a subdirectory `class-greetings`.

```
cd class-greetings
```

...and you'll see at least a greeting file from me in `smtuttle.txt`, and probably more, because each of you is now going to add a greeting file to this directory, also!

Notice that you haven't changed your cloned repository copy yet -- because you MIGHT be doing this problem at the same time as someone else, do:

```
git pull
```

...to make sure you have an up-to-date copy of the repository.

Then, in directory `class-greetings`, make a file *yourGitHubUsername*`.txt` that contains a greeting and your real name.

Stage this new file using:

```
git add yourGitHubUsername.txt
```

...and then commit it using:

```
git commit -m "adding yourGitHubUsername greeting"
```

Show that you've changed and committed your local nrs-projects `458hw02` repository by running the following commands -- run the following WHILE WITHIN your repository's `class-greetings` directory!! (You should then be creating the file to submit OUTSIDE of your repository...)

```
# make sure you run the following from the class-greetings directory
#     within your nrs-projects 458hw02 repository
echo yourName >> ../../458hw2-5-show-changed.txt
pwd >> ../../458hw2-5-show-changed.txt
ls >> ../../458hw2-5-show-changed.txt
git status >> ../../458hw2-5-show-changed.txt
```

Then -- cross your fingers and try pushing your updated repository back up to GitHub:

```
git push origin master
```

And, change your current directory to the one containing your `458hw02` repo:

```
cd ../..
```

...and submit your resulting file `458hw2-5-show-changed.txt` (which SHOULD be here if you followed the steps carefully/correctly).

(And I'll also look to see if your greeting file *yourGitHubUsername*`.txt` is successfully pushed to the GitHub version of `458hw02`.)