GitHub Setup Instructions for Local Installation on MacOS

Step 1: Setting up password-less SSH authentication

The basic idea here is that we want to set up a sort of “hand shaking” protocol between the local account on your laptop and the GitHub servers. This involves two parts: the first is to generate encryption files and a configuration file on your laptop, and then the second is to upload one of those encryption files to your GitHub account.

In what follows, I am going to assume that your local account has the username kara, and your GitHub account has the username ferner. So, wherever you see that in the instructions below, just replace either kara or ferner with whatever your local username and GitHub account username are, respectively.

A) On your laptop, in a terminal window, type:

cd ~/.ssh
ssh-keygen -t rsa

For the file in which to save the key, change it to: /Users/kara/.ssh/
id_rsa_github_ferner

For the remaining questions, just hit return.

This will generate two new files in the /Users/kara/.ssh directory:
id_rsa_github_ferner, and id_rsa_github_ferner.pub
The former is the PRIVATE key, that you should never ever give to anyone. The latter is the PUBLIC key, and is what you will be uploading to GitHub.

The final step here is to create a local configuration file. This file needs to be called config
If you already have a file called config in your .ssh directory, you can just add the following lines to it. If you do NOT have such a file, just create a new one.

Put the following lines in this file:

Host github.com
  Hostname github.com

```
  User ferner
  IdentityFile /Users/kara/.ssh/id_rsa_github_ferner
```

I am not sure how this note is going to be formatted, but the second through fourth lines need to be indented relative to the first line.

B) On your GitHub account in a browser:

Make sure you are logged into the GitHub account that you created this morning. In the top right corner of the screen, click on the pull-down menu next to your account icon, and go to Settings

Choose the tab on the left side of the Settings screen called 'SSH and GPG keys'

Click the green button under SSH keys to add a new SSH key.

The title can be anything you want. Then, on your laptop, open up the PUBLIC key file that you created in step A (id_rsa_github_ferner.pub), and cut and paste the entire key into the box that says Key on the GitHub website. Then, click on the green button to Add SSH Key.

Step 2: Cloning the repository on your laptop

Again, in a terminal window, do the following:

```
cd
mkdir math499
cd math499
git config --global user.name "Kara Ferner"
git config --global user.email "kara.ferner.17@cnu.edu"
```

(Note: you don't need to change the Kara or Ferner in these last two lines)

```
git clone git@github.com:ferner/phys441.git
```

This last command SHOULD download the repository! If it does not, then I have messed up in my instructions and have forgotten something. This is almost certainly what will happen the first time you try this. You can then think "Oh my, Dr. Brash is so dumb" and write to me about it and I can then try to figure out what it was that I missed.

cd phys441
git status

This command should report that everything is up-to-date and life is good.

git remote -v

This command should report the following:

origin git@github.com:ferner/phys441.git (fetch)
origin git@github.com:ferner/phys441.git (push)

Basically, what is is saying is that the GitHub origin of your local repository is as specified, which makes sense, right?

The final step is to tell your local GitHub repository that in fact, the origin repository is actually a fork of my master repository. This is important because in the future, that will allow you to update your forked repository with additional files that I create and changes that I make.

git remote add upstream git@github.com:brash99/phys441.git

(You don't need to change anything in that command!)

git remote -v

This command should report the following:

origin git@github.com:ferner/phys441.git (fetch)
origin git@github.com:ferner/phys441.git (push)
upstream git@github.com:brash99/phys441.git (fetch)
upstream git@github.com:brash99/phys441.git (push)