Assignment 1

Installing and Using Github

Requirements: None! Github is really easy to use and setup, so anyone with a working knowledge of computers can get started. If you have any questions along the way email me: Ibrahiem.mohammed96@gmail.com or send me a hangouts message, or if you're feeling self-sufficient, google it!

What is Github?

Github is a cool way to store all your code and projects on an online server so that you can access it anywhere. It also lets you keep track of changes and lets you collaborate quite easily. Its used a lot by many companies and colleges so its useful to learn it.

Installing Github

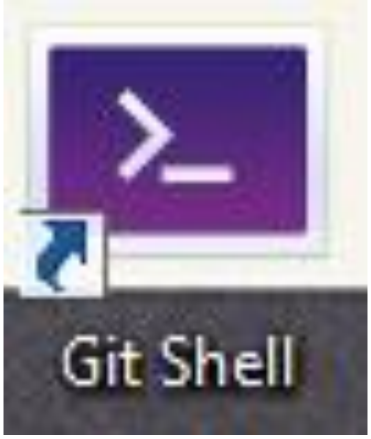
1) Firstly, go to https://github.com/ and make an account

2) Then, you need to get Github for Desktop installed
   a) https://desktop.github.com/

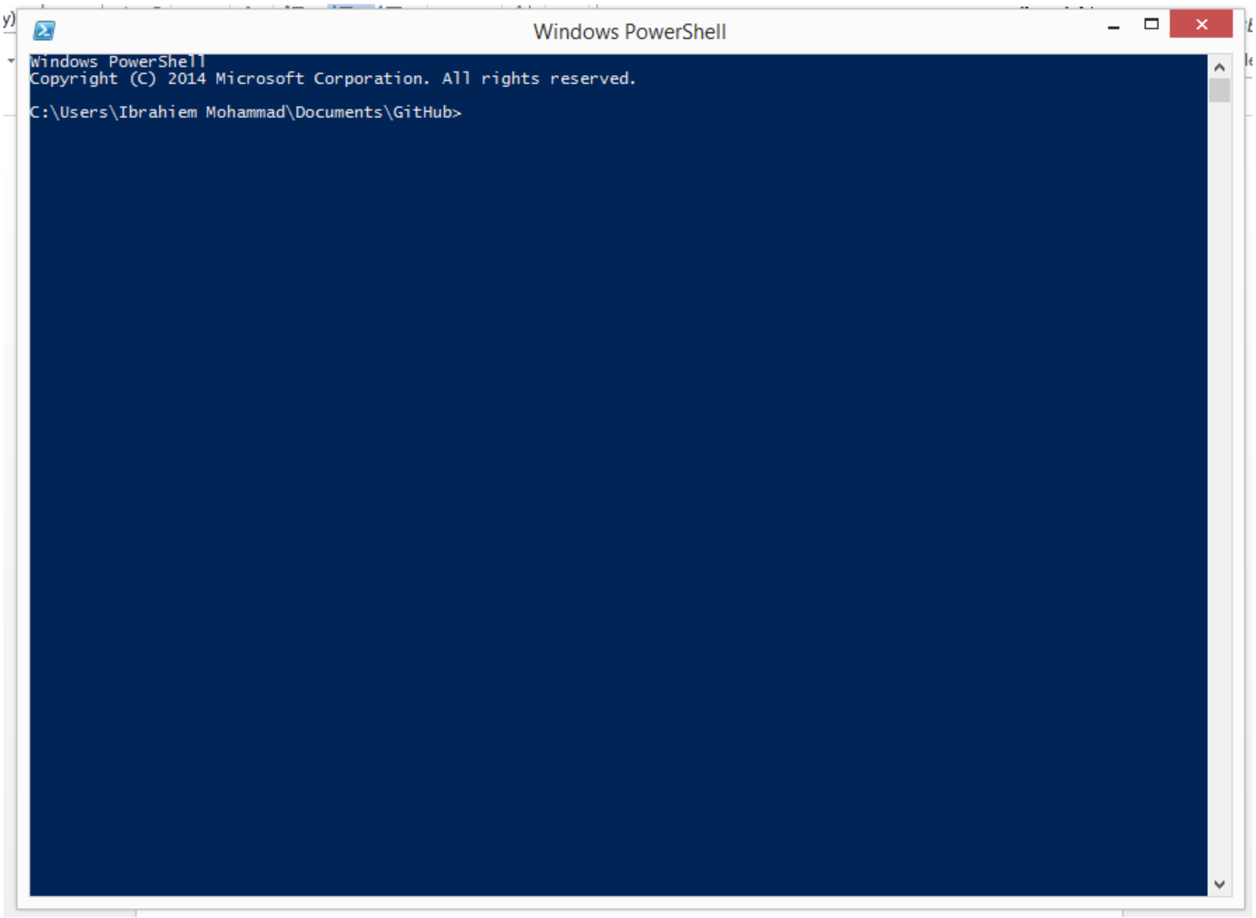
This will install Git Shell as well which you will use for your setup.

3) Next, open Git Shell

a) It will look something like this:



b) Opening it should give you this screen:



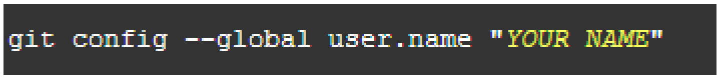
```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

C:\Users\Ibrahiem Mohammad\Documents\GitHub>
```

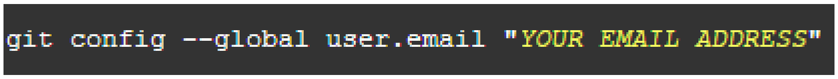
c) If you have gotten to this step, you’ve successfully installed Git!

d) The next step is to identify yourself to the Git Shell so that when you commit a project or file, Git will know who did it and help you keep track of the changes made by other people.

   i) Enter the following commands in the shell:



```
git config --global user.name "YOUR NAME"
```



```
git config --global user.email "YOUR EMAIL ADDRESS"
```

e) Congrats! Step one complete!

Authenticating Git

1) In order for us to be able to access the repository we must have the proper authentication. For our purposes, we do not need to set up a complex method of authentication, so we will use https cloning. However, I do recommend that if you have time to check out how to clone into a repository using an ssh key.

   a. **Extra Reading:** What is an ssh key? Think of an SSH key like a normal key. A normal key that opens a locked door. What is the door? In this case, that door is a secure server. So why do we need all this? We use ssh keys to make sure that only we have the access to the server. The disadvantage is that you can only access the files from the computer on which that ssh key is stored.

2) Before cloning into a repository and downloading it onto your computer, you will need to choose a pathfile for it.

a. Check the current pathfile by entering **pwd** into the shell.

b. I would suggest going into your Documents folder to make the repository but you can choose whatever folder you like.

   i. Some shell commands to help you navigate through the pathways:

      1. **ls** - see all files/folders displayed in current directory
      2. **cd ..** - go back to your previous directory
      3. **cd <name of directory>** - go into the desired directory (don't add the < > signs when entering this command).

c. Once you are in the directory/folder you want to set the repository in, move on to the next step

3) In order to authenticate or clone in to a repository you will need the link to repository:

   a. https://github.com/ibrahiem96/AP-Computer-Science.git
   b. Enter the following into your git shell:

      i. **git clone https://github.com/ibrahiem96/AP-Computer-Science.git**

      ii. Once you enter that the shell should ask for your username and password. Enter those and it should then download the repository onto your computer.

4) Congrats! Step 2 complete!

Using Git

1) Now comes the fun part.

2) First, in order to see if your local repository is in sync with the online you will need to perform a pull.

a) Enter **git pull** into the shell. The shell will say everything up to date (or it will download any changes made since after I wrote this).

b) Now each of you create a text file (you can use notepad, word, etc) which will have your name as the file name. Write something in the text file and save it in the repository (which is the folder that you cloned).

c) Once you've done that go to your git shell
   i) Enter **git status**
      (1) The shell will tell you that you have made changes that need to be pushed onto the server
   ii) Now enter **git add –A**
      (1) This will add all the changes you've made onto the 'stage' so you can commit them

   iii) Enter **git commit –m "enter a message here describing briefly what changed you've made"**

   iv) Finally enter **git push**
      (1) When you enter this command it may give you an error. If you get no error then your shell will tell you that your changes have been officially pushed and you are good to go. BUT if you do get an error:

         (a) Make sure to read the error carefully. It could be telling you that you either need to enter your password to authenticate the changes.

         (b) OR it could be telling you that there have been changes been made by someone else. In this case you need to enter **git pull**.

         (c) Enter **git push** again once you resolve the error and you should be good to go.

Making your branch

1) You're almost done with the assignment. The next step is creating your own branch so that you don't have overlapping changes which could cause conflicts and fatal errors.

2) Go onto your git shell and enter the following:

    a. **git checkout -b <your branch name>** *Remember don't add the < > signs. And use your name as your branch name.*

    b. now to make sure you've made your branch switch back to the master branch by entering: **git checkout master**

    c. now switch back: **git checkout <name of your branch>**

3) If you've gotten this far you're doing great. The last step is to make sure that I have access to the branch as well so I can access the work you guys put up in your branches.
    a. To do this enter the following:

        i. **git push origin <branch name>**

4) Once you've done this, create a text file in your branch (name it anything) and in the text file answer the following questions:
    a. Was setting up git easy/hard?
    b. What do you expect from this class?
    c. What do you like best about programming/computers/etc?
    d. What do you hate about it?

Okay, so now that all that is done with you are going to have to add, commit, and push the changes you made so that all changes are synced. If you forgot how to do any of those steps just scroll back up. A few reminders:

- Always do a **git status** check at the beginning and end of your session
- Always do a **git pull** before you push anything to make sure you are synced before you add anything else
- If you want/need more practice or want to learn more about Git visit this website: https://try.github.io/levels/1/challenges/1

The best way to know if you have mastered something is to teach it to someone else. So go ahead and teach your siblings/parents how to use git. Just for fun and some extra experience.