Homework 2
Computational Physics
Prof. Aldo Romero
Due September 10 at noon! (DUE THIS WEEK)

1) First, let us create a GIT repository. We will follow the class notes.

(a) Create a directory in which to practice via mkdir ~/Desktop/MyGITExample, then change to the new directory via cd ~/Desktop/MyGITExample.
(b) Configure git as follows. In the first line, use your name, and in the second, use your email address. You may have already done some of this if you were following along my lecture on git.

    git config --global user.name "Your Name"
    git config --global user.email "Your email address"
    git config --global core.editor vim
    git config –list

(c) Create an empty git repository in your directory: git init
(d) Create a file names.txt by revising the following command to use your WVUID, last name, and first name and then running the command (names.txt will be a sort of roster, containing people's names, contact information, and roles):
echo "boss, NetID, FamilyName, GivenName" > names.txt

    For example

    echo "Professor, 00000, Romero, Aldo" > names.txt

(e) Stage the new file for tracking: git add names.txt
(f) Check the status of your repository: git status. You should see names.txt
    in the output.
(g) Save a snapshot of your file; that is, commit your changes with a descriptive message: git commit -m "Create names.txt with a professor line."

2) Now, let's put your repository on GitHub.

(a) Access your GitHub account (whose account name we'll refer to as ID
(b) On GitHub, create a repository called MyGITExample. The new account dialog gives
    an opportunity, or you can use the "+ > New repository" menu in the upper right
    corner.

    i. Choose "Public", not "Private".
    ii. Leave "Add a README" unchecked.
    iii. Leave "Add .gitignore" unchecked.

iv. Leave "Choose a license" unchecked.
v. Click "Create repository."

(c)   In your MyGITExample directory, to tie your local repository to GitHub, run
git remote add origin git@github.com:GitID/MyGITExample.git
(after changing GitID to your GitHub ID).

Note: To make changes to origin after setting it, run
git remote set-url origin <URL>
where <URL> is the new URL you want to use.
(d) Run git remote -v to confirm that your remote has been set up correctly.
(e) Remember that you must use your token for the first time. Read the email I sent
     you guys.

3) Now, we're going to start collaborating through git. To do that, we need some
   collaborators. You will include myself (ahromero) and two of your classmates. You
   will need you GitHub ID of these people. Here is the list of all

Last Name, Name   email   GitHub user name

Acharya, Nischal,   na0090@mix.wvu.edu
Chafin, Sadie,   src0045@mix.wvu.edu
Moran, Michael,   mem0001@mix.wvu.edu   memoran1998
Mull, Nolan,   ncm0007@mix.wvu.edu
Powers, Jack,   jkp0011@mix.wvu.edu   Powers-Jack
Young, Morgan,   mey00001@mix.wvu.edu   myoung28

   Now go to settings/Manage access and include the names of the "collaborators" as
   follows:

your "employees" are the two students after you in the ordering (the first two students
alphabetically if you are last alphabetically) and add me (for a total of 3 people). Your
"boss" is the student before you in the ordering (the last student in the list if you are first
alphabetically).

4) Now, let's grant access to your collaborators.

(a) On GitHub, click on your MyGITExample repository, then "Settings" on the menu
under the repository name, and "Manage Access" in the left margin.
   •   Scroll down and click the green "Invite a collaborator" button.
   •   In the "Search by username, full name, or email" box, enter your three
   collaborators' GitHub IDs.

- Click "Add collaborator".

(b) If you do not have a GitHub username from your "employees" yet, just add the professor and those students have it. Don't forget to come back and add your missing "employee" later.

5) Once you have accepted access to the GitHub repo belonging to your "boss", it's time to act as an "employee" and contribute to the repository of your "boss" by making a one-line addition:

(a) Copy your boss's repository from GitHub into a directory called gitBoss: git clone git@github.com:bossID/ MyGITExample.git ~/Desktop/gitBoss (where bossID is your boss's GitHub ID)

(b) Change to your "boss" repository directory: cd ~/Desktop/gitBoss
(c) Run git status to verify that names.txt is in the repository.
(d) Run
echo "employee,NetID,FamilyName,GivenName" >> names.txt
(After replacing NetID with your NetID) to append a line of information to the roster information. **Note that we have two greater-than symbols** instead of the one that we have seen for redirects before. This means to append, i.e., add a line, to the file names.txt, instead of overwriting the file.
(e) Add the changed file to the (boss) stage: git add names.txt
(f) Commit the change to the local (boss) repository:
git commit -m "Add a line to boss's file."
(g) Pull the GitHub repository of your "boss":   git pull origin master and fix conflicts if necessary (there should not be any conflicts).
(h) Push your changes to the repository: git push origin master.

6) Okay, back to acting like a "boss".

(a) Return to your first repository: cd ~/Desktop/ MyGITExample
(b) Update your local repository to receive your employee's change: git pull origin master. Probably you need to do this twice, as you have two "employees".
(c) Check your names.txt file to see that is has two/three lines, your boss line and your employee's lines (two additional). If it isn't there, that's okay, check back again soon.
(d) Add a header line to names.txt by adding a line at the beginning of the file of the form role,NetID,FamilyName,GivenName. This time, NetID should be written literally. Do not substitute your NetID.
(f) Add your changes, commit them with a descriptive message and push them to your repository.

What do you need to submit?

Nothing. Just send me an email when this is ready but before the dateline. I can then clone the repository and check that the info is correct.