ECE884
Assignment Submission Instructions

Introduction:
We will be using a *version control system* called **git** for assignment/project submission and possibly grading. **git**, and similar version control software, are common standard tools used by computing professionals and engineers. A basic working knowledge of **git** will be helpful/necessary for the rest of the class. The following list of references provide tutorials/guides about **git**:

- http://git.huit.harvard.edu/guide/
- Software Carpentry's Git Lesson
- Git Immersion (A git tutorial)
- git - the simple guide
- Atlassian's Git Tutorials

Git is installed on the MSU high performance computing center (HPCC) and the MSU Gitlab. For those interested in installing git on their personal systems (laptop, PC, MAC), git can be downloaded from here. For Windows' users you may choose git bash, from here, which makes the command line similar to a Unix system.

The git repositories used in this class will be hosted on the MSU Gitlab server. Groups and repositories will be setup for enrolled students in the group ece884fs2022. Each student will have access to two repositories:

instructor

and

username,

where *username* corresponds to each student's MSU NetID. The ece884fs2022/instructor repository will be used for distributing assignment instructions and template source code files for assignments/projects, and thus it will be read-only (i.e., you cannot **commit**/**push** files to this repository but you can **pull** files. See command examples below!). The ece884fs2022/*username* repository will be used by students, individually, for submitting their assignments/projects.

Setup:
First, you will set up SSH keys to enable secure and password-free access to the Gitlab server. (This assumes that you have NOT set SSH on Gitlab before.) The following instructions illustrate the steps on linux and unix-based systems (e.g., MacOS, or Windows using git bash terminal). Full instructions can be found here.

Briefly, you should:

0) Install git or git bash on your local device (follow the links above).

1) Activate your account with the MSU gitlab cloud.

Then,

1. Log onto a system with git installed (preferably, your own preferred local device)

2. Generate a key (assuming you haven't already done so) by typing:

```
ssh-keygen -t rsa -C "your_email@msu.edu"
```

(You should keep hitting ENTER to accept the defaults.)

3. Copy the contents of the public key file ~/.ssh/id_rsa.pub
4. Log into the Gitlab web interface here using your MSU NetID credentials
5. Click on Settings drop down menu (it is a gear icon in the upper-right corner of the screen) and select Profile Settings
6. Click on "SSH keys" (from the tabs on the top of the screen.)
7. Paste your public key in the Key form
8. Name this SSH key. For example, "MyPC1".
9. Click on "Add Key" to indicate you are finished.

Next, go back to the terminal on the (local) device with git installed, create a folder for the class, then from this folder run the following commands to clone the repositories (due to MSU's SSH access restrictions, you may have to do this while on the campus network. Alternatively, login thru **vpn.msu.edu** via, e.g., the **PulseSecure** app):

```
git clone git@gitlab.msu.edu:ece884fs2022/instructor.git
git clone git@gitlab.msu.edu:ece884fs2022/username.git
```

where *username* should be substituted with your MSU NetID. Notice that you should now have two directories, instructor and *username*, on your local device which contain the contents of each of the repositories.

Lastly, you will want to configure git on the system/device you are using with your personal settings. The following commands set values for all your repositories:

```
git config --global user.name "YOUR NAME"
git config --global user.email "YOUR EMAIL ADDRESS"
git config --global core.editor "Text_Editor_Name"
```

where Text_Editor_Name may e.g., be “nano, vi, vim, …”

Accessing the Assignments:
Each assignment in the future will be uploaded to the ece884fs2022/instructor repository as a new subdirectory. For example, the first assignment will be placed in ./instructor/project/1/, the second assignment will be placed in ./instructor/project/2/, etc.

To access the assignment files, go to your local copy of the instructor repository, and run the “git pull” command:

```
cd instructor
git pull
```

After this command is completed, you will see the updated directory structure for the instructor repo. For a potential autograder to work properly, it is crucial that you create the same directory structure within your own (local) repository (i.e., *username*). The following commands demonstrate how to do this.

```
cd username
mkdir -p project/N  (replace N with the project number)
cp –r ./instructor/project/N/   ./project/N/ (change the path to the local./instructor repo, if
```

necessary)

Submission:
For each of the projects, you will ***add*** and ***commit*** your files to your local *username* repository and ***push*** them to the Gitlab server for grading. The following commands illustrate how to do this.

On your local machine:

```
cd username
# add files for your submission
git add project/N/code.py project/N/Project_solution.pdf
git commit -m “Project N solution.”
git push
```

Note: please adhere to the naming conventions given for files and directories in these instructions when submitting projects/assignments. In particular, these are case-sensitive and exact in the naming – if the incorrect names are used, any potential grading script would fail to recognize your submission, and the resultant grade for the submission will be 0 points.

If you run into difficulties during this process, you can ask for help from MSU gitlab support and/or through posting on our Piaza class group.

References:

MSU Gitlab website: gitlab.msu.edu