CS3240, Lab 5: Pre-Reading Version Control with git

The goals of these lab activities are to learn the basics of using git and GitHub together.

Before Coming to the Lab Section meeting: Before coming to lab, you must do the following things to prepare. There are readings and also installing software on your laptop.

1. Install git on your laptop. See instructions below.
2. Create an account on github.com if you do not already have one. Use your UVa email ID, which will allow you to request an educational discount. Click on *Request a Discount* on that page. This will allow you to have a number of private repositories (5, in fact). Later this term, one person on your project team will need to use one of these private repositories for your team project.
    - **Note:** Request a Discount as soon as possible. In past terms, GitHub sometimes has taken many weeks to process these. For today's lab, it doesn't matter, but you need this for when the project starts.
3. Download a copy of the Git Cheatsheet from GitHub's training page. You'll want to refer to this during lab, so you may want to print a copy and bring it to lab. Otherwise be prepared to look at it on your computer.
4. Look over the two suggested readings on git:
    - "Git Immersion" http://gitimmersion.com/
    - Simple tutorial of the basics: http://rogerdudler.github.io/git-guide/
5. Complete the pre-lab activity listed below and submit the screen-capture described there to Collab before your lab section begins.

During the Lab Section meeting: You'll carry out various coding activities (to be described later).

Updates, Issues, Gotchas:

1. None yet, but watch this space.

Installing git on your Laptop

Linux and OS X users may have git installed already (type *git* to see). If not, if you are using a package manager (apt-get, brew, etc) feel free to use that to download git if you know how to use them for this. For everyone else....

You can download a version of git from GitHub and other providers, but we recommend you download it from the main git site at this URL: http://www.git-scm.com/downloads
Note: This is particularly important for Windows users, since this download comes with a bash-shell (like you find on Linux) called **git-bash** that we ask you to use. We've seen git commands fail mysteriously when run from the Windows command-prompt. Don't use that! Use git-bash! It's a bash shell command line environment, just like Linux, with git commands and also basic Unix commands installed, plus some other useful things in its /bin folder: ssh, scp, perl, curl, vi, vim. If

you need to do UNIX shell kinds of things on your Windows machine, this is a nice thing to have. (FYI, if you know cygwin, it's like that but lighter-weight, since it installs less.)

Follow the directions to install git. Here are some notes for Windows users. When running the installer, I kept all the defaults, except you need to make sure you choose the following:

1. In the window *Select Components*, I chose
    - Under *Additional Icons*, I checked *On the Desktop* (This puts the Git Bash icon on your desktop.)
    - Under *Windows Explorer* integration, I checked *Get Bash Here* (This lets your right-click on a folder and open Git Bash in that folder.)
2. In the window *Adjusting your PATH environment*, choose *Use Git from Git Bash only* (Experience in previous classes suggests that you should only use Git from Git Bash.)
3. In the window *Configuring the line ending conversions*, keep the default.

Initial configuration (one time only, for all platforms):

Start a shell window to use Git (on Windows use Git Bash, on Linux or OS X open a "terminal" window), in that shell window, set some configuration values. Type the following but update the final arguments:
git config --global user.name "Your Name"
git config --global user.email "your_email@whatever.com"

There are times a git command will bring up an editor window. The default is *vi*, but if you want *emacs* instead, type:
git config --global core.editor emacs

Pre-lab Activity and Submission to Do *Before* Lab

The following steps must be done before coming to lab. By doing these steps, we'll know you've got git installed correctly and have created a GitHub account. There's a submission to be done before coming to lab to show that you've done this. If you don't complete this, some points will be deducted for lab participation this week.

First, create a repository on GitHub:

1. Sign into www.github.com with your account. (Don't have one? See the top of this page.)
2. Create new repository. (See instructions they give, or just go for it by clicking the plus-sign.)
3. Name the repo *cs3240-labdemo*, and make it public.
4. Check the box to initialize this with a README file.
5. Add a .gitignore for Python, and choose the MIT License.
6. Now click Create Repository.

GitHub will show the main page for your new repo.
Near the bottom-right, you'll see a text-box with the text *HTTPS clone URL* right above it, and an icon like a clipboard with an arrow to its right. This textbox contains the URL that you can use with a git-command on your local machine to clone the directory. To do that:

1. Click on the clipboard icon to copy the URL to your clipboard.
2. Go to your shell window on your local machine, and *cd* to a folder where you want to keep your projects in folders. (Create such a folder if you don't have one.)
3. Type *git clone* and then paste in the URL from your clipboard and hit return. It should look something like this:
   git clone https://github.com/hortonuva/cs3240-demo.git
   (Windows users: in Git-Bash, you might find copy-and-paste commands by right-clicking in your window. If not, see this link for some more info on how to cut and paste.)
4. This will download all the files from the remote repo into a directory (named after your repo) in the current directory.

Congrats! You now have a local repository on your machine you can work with, plus a separate repository on GitHub where you can upload your files for safekeeping or to share. Now do the following from the shell window on your local machine:

1. Use *cd* to make sure you're in the *cs3240-labdemo* directory.
2. Type *ls -a* to see what files are there. Note the hidden files that begin with a "."
3. Create a file *hello.py* that prints "hello". Make sure it runs under Python 3. (Use vi or emacs to create the hello.py file)
4. Type *git status* and make sure you understand what that tells you.
5. Type *git add hello.py* to add that file to the staging area.
6. Type *git status* and make sure you understand what that tells you.
7. Add the file to your local repo by typing *git commit -m "Initial commit"*
8. Synchronize your local repository with your remote repo on GitHub by typing *git push*

Go back to your web-browser showing your GitHub repo. Refresh your browser window, and you should see the hello.py file in the list of files.

To submit before lab:

Take a screen-capture of this window, and submit the image file to *Lab 5 Pre-lab* under *Assignments* in Collab before coming to lab. Make sure your image file shows your repo name (which includes your GitHub id) and the list of files. See below for the kind of screen-capture I want.

Lab Activities:

(Activities will be posted before lab.)

Example of screenshot you need to submit to Lab 5 pre-lab assignment