CSCE 156 – SVN and Github Repositories

“Hey, that code you demonstrated in class was awesome! I wish I had a copy of it!”, “I really don’t know how to get started; I’d love to have a good example sitting right in front of me in Eclipse!”, “I dozed off and completely missed all that code you wrote and now I’m screwed!”. Have you felt yourself thinking or saying anything like this? Have you thought about saying any of them, or just said that you thought about them? Wish you had the code that I produced without the unbearable tediousness of unzipping an archive file? Now you can, because I’ve made all the demo code available in an SVN (Subversion) repository as well as on Github for easy checkout.

SVN
Apache Subversion (SVN) is a software versioning and revision control system in by many software development projects. It allows teams of developers to access shared code and collaborate on changes by checking in/out new versions. For more details, see:

- [http://subversion.tigris.org/](http://subversion.tigris.org/)

To access this course’s repository, you will need to install an SVN client to “check out” the code and any changes that may be made. This document describes one way of doing this through Eclipse. If you use a different IDE or prefer a different SVN client, that is perfectly fine, but you will need to seek alternate help.

Instructions
Various versions of Eclipse may differ slightly, so you may want to find a more current SVN/Eclipse tutorial if this does not work for your version.

1. Be sure that you have an SVN client plug-in installed in your Eclipse. My suggestion is to install Subclipse. The tutorial listed above as directions for and older version of Eclipse. With newer versions, there is now an “Eclipse Marketplace” that you can automatically install plugins and dependencies. To do this:
   a. Go to Help -> Eclipse Marketplace
   b. Search for “Subclipse”
   c. Hit the... Install button!
   d. Note: you may also need to install SvnAdmin
2. Once installed and restarted, right-click in the Project Explorer area and select Import...
   a. Select SVN -> Checkout project from SVN
   b. Create a new repository location using the following URL: [https://cse.unl.edu/svn/cbourke/CSCE156](https://cse.unl.edu/svn/cbourke/CSCE156)
   c. Use your cse login and cse password to login
3. Checkout the CSCE 156 project. You will have read-only access so you won’t be able to check anything in, but to get any changes that I check in, you can “synchronize” your project by right-clicking it, select Team -> Synchronize with Repository and Updating any files that have been changed.

If changes are made or code is added you will need to synchronize your local copy with the repository. To synchronize, do the following.

1. Right click your project
2. Select Team > Synchronize with Repository
3. When switched to Team Synchronizing view, right-click your project and select Update
Creating your own SVN
If you choose to work in pairs for your project work, then you are highly encouraged to collaborate by creating an SVN to store your code. Then you can both check in and check out code from each other. Moreover, your SVN repository will keep a safe backup of your code! See the following for instructions on creating your own private SVN:

http://cse-wiki.unl.edu/wiki/index.php/Setting_up_an_end_user_SVN_Repository

Git
The code is also hosted on Github at the following URL.

https://github.com/cbourke/CSCE156

Git is a newer, distributed revision control system that has become very popular. The learning curve can be a bit steep as most of the interaction is via command line arguments. Here are a few resources to help you out.

- https://github.com/
- https://help.github.com/articles/set-up-git/
- https://github.com/quinnliu/gitCommands

You are also encouraged to collaborate with your group by creating a git repository and sharing it. Be aware, however, that unless you have a full account (free for students), you cannot make your repository private. You will need to make sure that the code for this course is kept private to conform to the Academic Integrity policy of this course and the department. Alternatively, UNL hosts its own git server at the following URL.

https://git.unl.edu