

CECS 282 - Homework 0 - Getting Started with GitHub

Git is a tool used in the software industry for “revision control”: allowing multiple people to contribute changes to a set of project files, while managing conflicts in changes, allowing reverts to previous versions, and generally hosting and coordinating all files (called a “**repository**”) related to the development of a software project. Git draws inspiration from older software like Subversion (“SVN”) and Concurrent Versions System (“CVS”). As a professional developer, you will be expected to know how to use at least one revision control system, and in that spirit we will use Git this semester to distribute lecture, lab, and project files.

GitHub is a website that uses Git and allows the public to host public code projects for free. Many sites use Git like this, and GitHub is the largest of the bunch, home to the largest collection of FOSS (“free open source software”) projects on the Web. Anyone can make a free account and publish their code to GitHub’s website for others to see and use.

Assignment

Getting Started:

Install a Git client on your computer (<https://git-scm.com/>). If using Windows, you should choose the option “Git from the command line and also from 3rd-party software” when asked. You can also choose to install a GUI client, but the instructions here will be performed on a command line / terminal.

Your First Clone:

Create a folder on your computer where you would like to save the course lecture notes – put it somewhere you can find easily. Then open a command prompt (on Windows) or terminal (Mac/Linux). You need to change the working directory of the prompt to the folder you just created.

- On Windows, you can highlight and copy your folder’s **path** by clicking the address bar in Windows Explorer and pressing Control+C. You can then paste that into the command prompt after typing **cd** to change the working directory.

Once you are in the right folder, run the command

```
git clone https://github.com/terrell-cecs282/Lectures.git
```

which will make a **clone** (duplicate) of the course lecture notes on your computer.

Exploring the Notes:

Feel free to poke around the Lectures folder. The notes are broken into folders, and each folder contains a number of Visual Studio “Projects”, with each project having one or more C++ code files. If you aren’t using Visual Studio, you will be able to make your own “Project” equivalents and copy my source files to your editor.

In The Future

The Lectures repository only has a few files at the moment – I’ll be making lots of additions and changes as the semester goes on. But I can only **push** changes to *my* copy of the repository on GitHub; the clone that is on your computer will be unaffected by my changes. To receive my changes, you’ll need to run **git pull** from a command prompt in your local folder to **pull** any

changes that exist on the GitHub **origin** copy of the repository. Do this once a week before lecture begins to make sure your notes are always up to date!

Making Your Own Changes:

You can edit the lecture notes to try things on your own, but doing so may bring you into **conflict** with changes that I push to the origin repository. If this happens, you will receive a message about **merge conflicts** when running `git pull`. For this class, the easiest resolution is to:

1. Make a note of which files are in conflict, in particular, `.cpp` and `.h` files.
2. Abort the merge: `git merge --abort`, which will restore your files to the changes you made.
3. Make a backup of the conflicting files **if you would really like to keep them**.
4. Reset your repository to unmodified files: `git reset --hard HEAD`
5. Run `git pull` again.
6. Roar in triumph.