# GIT Assignment

### **Overview**

The purpose of this assignment is to get you set up for the course, and get started with git and GitHub. Git is a version control system (used to organize source code when multiple people are working on it), and GitHub is a hosting service for organizing git-controlled projects.

This assignment will walk you through setting up git and GitHub for use in this course, give you a chance to use an online tutorial (you'll probably do that a lot in this course), and give you some practice making and submitting changes to a repository.

#### ***Assignment Objectives***

* Get setup with git, and GitHub
* Learn the basics of using git
* Be able check out and submit changes to a git repository
* Evaluate an online programming tutorial

### **Assignment Details**

There are lots of pieces to this assignment (as you need to get setup with everything), but all of them should be straightforward. These will take some time--*do not leave this until the last minute!*

#### ***Configure Git and GitHub***

* Sign up for a GitHub account at <https://github.com/join>
* If you already have a user account on github use that.
* That way he can add you to our GitHib organization and give you access to the repository.
  + Configure git so that it knows who you are, so that in a multi-person project each commit is tied to a specific person. In the **console** (with git installed), run the following commands:  
    git config --global user.name "YOUR NAME HERE"
  + git config --global user.email "YOUR UPS EMAIL HERE"
* You'll also want to [set up an SSH key](https://help.github.com/articles/generating-ssh-keys) to allow you to push and pull over SSH (the Secure Shell protocol--this allows you to connect to GitHub securely).
* You can also setup git to use a graphical text edit (like Sublime) for writing commit messages using git config --global core.editor <command>

#### ***Git Tutorials***

At this point, you should be all set up to use git! But how exactly do you use this thing?

While we'll be going over the basics in class, another good way of learning a new programming language or tool is to take advantage of the many online tutorials and guides.

There are numerous tutorials for git available; for this assignment, you will be using and evaluating one of the online, interactive versions:

* If your last name begins with the letters **A-E**, [Git Immersion](http://gitimmersion.com/lab_01.html) (labs 1-35)
* If your last name begins with the letters **F-M**, [Learn Git Branching](http://pcottle.github.io/learnGitBranching/) (all "main" tutorials)
* If your last name begins with the letters **N-Z**, [tryGit](https://try.github.io/) (challenges 1-25)

You are welcome and encouraged to look at the other tutorials--particularly if your assigned tutorial isn't helpful for you. *But you still need to complete your assigned tutorial so you can evaluate it!*.

* There are lots of other guides and resources out there as well. A couple of recommended ones:
  + <http://betterexplained.com/articles/a-visual-guide-to-version-control/> Note: This is a GREAT read for understanding version control in general! Be sure and read the followup article on Distributed Version Control
  + <http://git-scm.com/book/en/Git-Basics>
  + <http://rogerdudler.github.io/git-guide/>

#### ***Committing a Change***

Now that you know how to use Git, let's do something with it. For this part of the assignment, you will **clone** an existing repository, **modify some files**, and then push out your changes.

1. Start by cloning the git-assignment repository (**Note:** all instructions in this assignment describe cloning via HTTPS. You are welcome to clone via SSH instead; generally you replace 'https://github.com/' with 'git@github.com:')  
   git clone https://github.com/himsx/git-assignment.git
2. Switch into the git-assignment directory  
   cd git-assignment
3. Create and switch to a new branch to do "development" on  
   git checkout -b develop  
   It may feel a little silly to make a separate branch for the small number of changes we're making, but it's good practice!

* Modify the class\_list.txt file to include your name and information about yourself, following the format ou may want to backup your changes at this point by committing them  
  git add classlist.txt

1. git commit -m "<a descriptive message of your commit>"
   * **Remember**: commit messages should summarize what the commit does to the repo in less than 50 characters (you can add more detail on the following line). Use the imperative tense: "adds Jared’s name to the list"
2. Modify the tutorials.md file to add a review of the git tutorial you completed. This does not need to be extensive, but should include at least one thing that "works" about the tutorial and one thing that doesn't. You should also look through other people's comments, and expand onto them if you can. Our goal is to make a nice summary of these tutorials that may be helpful to others (e.g., future students)!
   * This is the most involved part of this assignment.
   * Yes, using a Wiki might be more effective at achieving this goal, but we're learning git.
   * Similarly, you should make these changes using the command-line interface, rather than simply editing through GitHub.
   * Note that this file is formatted using [Markdown](http://en.wikipedia.org/wiki/Markdown)
3. Commit your changes to your local repository (see above)
   * Merge your development branch back into your master branch. First checkout the master, then perform the merge.  
     git checkout master
   * git merge develop
     1. Before you push your changes back out to the rest of the repository
   * Finally, you should push your changes back out to the rest of the repository. First, make sure to pull in anything that has changed since you began editing  
     git pull  
     and finally push the changes  
     git push
4. If you get any merge conflicts throughout this process, you'll need to resolve them!

And that does it for this part of the assignment!

### **Submitting**

Successfully committing and pushing the required changes to the git-assignment repo GitHub counts as a submission of this assignment. Remember to **push** your changes to GitHub!