This is information for using git via **command line**. You can use a GUI, but to me the GUI doesn’t give as much control and doesn’t make what’s happening very clear.

A lot of this information on what to do is given by running a ‘git status’. I’m providing this for a git workflow process.

<http://pcottle.github.io/learnGitBranching/> - nice visual walkthrough

**How to navigate to working directory**

**mac:**

1. open terminal
2. ‘cd’ to the project directory (csce4444)

**windows:**

if you installed GitHub Desktop gui, you should have a Git Bash program. install the GUI if you haven’t.

1. open ‘Git Bash’
2. go to mac step #2

**Before you start making changes**

1. Run ‘git pull’ to make sure your local branch is up to date with the remote branch
   1. If you have changes already, keep working and commit.
   2. Run ‘git pull’

**Before you commit/push**

1. Don’t push until you think your code is working fully. committing is okay to do more often.
2. Run ‘git diff’ to verify the changes you made are what you want to commit
3. Run ‘git status’ to view changed files
4. Run ‘git add FILENAME’ for each file you want to be added to the commit
5. Run ‘git commit -m “MESSAGE”’ to commit your changes with a message (commits are local)
   1. think of commits as the Save from a Word doc. you don’t actually change anything unless you save.
6. Run ‘git push origin/BRANCH’ to push your changes to the remote repository

**Adding a new file (untracked)**

1. If you run ‘git status’ and it shows untracked files, those files are not associated with your git repository. They are just there.
2. To add the files, run ‘git add FILENAME’

**To discard changes in your local repository (not staged for commit)**

1. Run ‘git checkout FILENAME’

**To discard changes in your local repository (staged for commit)**

1. Run ‘git reset HEAD FILENAME’