Getting Started with GIT

CSCI 2300

What is GIT

- Version control software
- Used for collaborations
- Also useful for individual projects

GIT repositories for this class

- Everyone in this class has a GIT repo created on the department server
- Should have received an email about this
- Follow the instructions in the email to set up your password
- git.cs.slu.edu
- Everything you "push" to the git repo is stored on the department server.

Cloning your repo

- Log in to a lab computer or to hopper.slu.edu
- Create a "repos" directory under your "home" directory on the Linux system:
  `mkdir repos`
- Navigate to the repos directory:
  `cd repos`
- Create csci2300 directory:
  `mkdir csci2300`
- Navigate to csci2300 directory:
  `cd csci2300`
- Clone your git repo for this class:
  `git clone <your repo>`
  `git clone git@git.cs.slu.edu:courses/fall18/csci_2300/instructor.git`

Git Workflow for this class

- Git operates on branches
- The main and default branch is 'master'. See all branches with:
  `git branch`
- Branch with a * shows the current branch
- Only working and completed code should go into 'master' branch
- Create a feature branch when working on assignments
  `git branch homework1`
- Switch to homework1 branch:
  `git checkout homework1`
- Multiple feature branches can exist

Add files to your branch

- Create files and directories, as you normally would:
- Add the files to the 'staging area'
  `git add <file_name or dir_name>`
- Look at the state of your branch:
  `git status`
- Commit files to your branch
  `git commit [file_name]-m "Commit description"`
  - If `file_name` is not provided, everything will be committed
  - If `-m` is not provided, a default editor will open and force you to write a commit description. If you don't write one, your code will not be committed
- Look at the state of your branch now.

Merging with 'master'

- Check out the 'master' branch:
  `git checkut master`
- Merge your feature branch:
  `git merge homework1`
- Push your code to the git server:
  `git push`

Your OFFICIAL submission is what you pushed in master branch

Useful Tips

- Do frequent commits in the feature branch
- If you mess up, most likely there is a way to fix it
- Avoid multiple 'clones' of your repo