INFO2180 Tutorial 2

In this tutorial you will go through the basics of using Git.

Git is a version control system for tracking changes in computer files and coordinating work on those files among a group of people. It is usually used for source code management in software development projects, but it can be used to keep track of changes in any set of files.

As you go through this course you will be using Git and Github with your programming projects. This will give you good practice not just for this course but also for when you go out into the working world as many organisations are using version control for source code management.

Go through the following Git Tutorial to learn the basics of using Git and Github pages:

- Git Handbook: https://guides.github.com/introduction/git-handbook/
- A Guide to Using Github Pages - https://www.thinkful.com/learn/a-guide-to-using-github-pages/

Discussion Questions

1. If you are not currently using a version control system like Git how do you currently track changes to your source code?
2. When you are working on projects with multiple persons, how do you currently ensure that everyone has the latest version of your code?
3. What if someone makes a mistake and introduces a bug in your code, how are you are able to go back to a previous working version?

4. How can a version control system like Git help to avoid these and other problems?
5. Do you know of any services online where you can remotely host your Git repositories?
6. What are some git commands that you know of and what do they do?
7. To *add* changes to your files to the Staging Area in Git what command would you use?
8. If you then want to store your staged changes to your Git repository what command would you use?
9. What if you wanted to undo changes that you have made to a file that was staged in your repository?
10. When developers are working on a new feature or bug they want to fix they'll often create a copy or *branch* of their code they can make separate commits to. How is this done in Git?
11. Once you or a team member are satisfied that your new feature is working or bug is fixed how do you *merge* those changes?
12. Changes that are committed to your repository are done locally (on your personal computer) first. How would we now *push* these changes to a remote repository (e.g. on Github) so that our other team members can also see and get our changes?

The following resources will be helpful:

- Installing Git - https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- Git Handbook - https://guides.github.com/introduction/git-handbook/
- Git Cheatsheet - https://services.github.com/on-demand/downloads/github-git-cheat-sheet/
- Learn Git - https://www.codecademy.com/learn/learn-git
- Github Pages - https://pages.github.com/

- A Guide to Using Github Pages - https://www.thinkful.com/learn/a-guide-to-using-github-pages/
- Github Desktop GUI - https://desktop.github.com/

At the end of this tutorial you should know the following Git commands:

- `git init`
- `git status`
- `git add`
- `git commit -m`
- `git log`
- `git checkout`
- `git reset`
- `git branch`
- `git merge`
- `git remote`
- `git pull`
- `git push`

As well as how to use Github Pages.