Week 1 Day 1 Notes:

* **Welcome**

Will post note layout at beginning of class

Be sure to be there at the beginning

* **Syllabus Review**

Need to take notes for attendance

No late work allowed (Unless serious conditions.)

Unique code required.

2 exams, Mid and Final

* **Understanding Github and Github classroom**

Github helps us keep track of code and changes, Github Classroom also allows the instructor to keep track of things and add notes/changes as well.

Every assignment that requires code will have its own unique github code. The link creates a repository and gives access to the instructor, he cannot access your other repositories.

Create a professional profile for Git, you can use this for future work and have your work from class to show.

Good to have your code stored somewhere, Github is a good place to do so as it is easily accessible and shareable

*Version Control System* (VCS)- A way to manage files and directories, track changes over time, and recall previous versions. Source control is a subset of VCS.

If you push a bad version, you can always easily revert back to a working version with a VCS.

*Git* - created by Linus Torvalds, april 2005

Cross platform

Used by almost every company, very important to be familiar with

Git Distributed Version Control

*Repository* (Repos)

Good for organizing a single project, it can contain a wide variety of data constructs.

*Branches* allow you to switch between different versions. This way, you can easily test something, and if it doesn’t work you can easily go back to your latest working version before you begin experimenting.

*Main branch* has the latest working commit, you probably wanna branch off before you start experimenting so you can easily fix any mistakes.

Important to make branches off main when working on things, so you know your change will work on the main version.

*Dev branch* is what you call the branch you are working through solutions on.

Anytime you make a change to a file, that is a commit. You can see the history of commits, making it easy for you and others to see what was being done.

There are commit messages, be sure to write informative commit messages. This will make fixing bugs much easier.

*Pull Request* - Initiate discussion about a commit, good when working in a team. In this class, help requests on code must be done through pull requests.

*Merge and Deploy* - once you fix issues, you can merge it into the branch. Merge conflicts must be worked out with the team, pull requests preserve a record of the

historical changes to your code.

*GitHub* - Free, most popular Git host, allows online collaborations from anywhere.

Forking is another term for branching

Sometimes developers choose to place repe on GitHub

\*Lab 1, do not merge dev branch