**Lab-0 Week 1**

In this lab we have gone through the basics of how to setup, install and utilize Git and GitHub to keep track and organize our programs as individuals and avoid tripping over each other in a dev team.

**We have learned:**

**Creating a repository in GitHub**

-GitHub is an important tool for any software developer. It allows users to make online backups of their code that they can revert to at any time and access anywhere on any machine. GitHub allows users to share their work with others and collaborate in groups of virtually unlimited size seamlessly and simultaneously.

**Using VS code with Git and cloning a repository onto a local machine**

-VS Code is a powerful software development tool with many useful addons. In this lab we have learned how use the console using GitBash to interface with GitHub and “clone” our online repositories to our local machine.

**Branching**

-We learned how to use Git to create “branches” of our work. This allows us to update or make changes to our program on a separate but simultaneous version of our files without effecting the original or “master” files until we are ready to add the changes.

**Committing**

-Committing is one of the most important steps to using GitHub. Committing your code is effectively like creating a new version or chapter of it. This helps developers keep track and separate of all their changes so they can more easily identify where something has gone wrong or where a bug has appeared.

**Merging**

**-**Merging is a crucial part of committing and branching. Merging allows developers to combine multiple versions/branches/commits of a file back into one unified version/branch/commit without having to manually get each file and copy new code over.

**Pushing**

-Pushing allows developers to add new or merged files, commits or branches back into the online repository.