**Github**

Github is a tool that is used as a version control system for software development. Git is an open source version control system. This system allows your projects to have multiple branches and has the ability to merge these branches together. Github allows you to connect your Git repositories to the cloud. You can use Github through the command line or use Github Desktop to use Github. This reference guide will be using Github Desktop

**Getting Started**

To start you need to create a **repository**. There are multiple ways to do this. One of the easiest ways is to create the repository on Github.com. On the main page there is a button titled “New” which allows you to create a repository. The next step is to be able to work on the repository on your local machine. You can click the “Clone Repository” button on Github Desktop to create the repository on your local machine. One important thing to note is that the local repository and remote repository are connected but not the same. There are Github commands to move code from one to the other.

**Github Keywords**

*Repository –* The collection of branches that the project is defined on. Repository can be local(on computer) or on web(Github.com) and the local is often connected to the web

*Branch –* A different version of your project file. Can be merged or discarded.

*Checkout –* Switch working branch to another branch

*Merging –* Apply the differences in one branch to another(typically Main)

*Main –* The branch that other branches merge into

*Stage –* Put your changes on your current branch into staging

*Commit –* Apply all staged changes to the current branch

*Publish –* Create the new local branch on the remote repository

*Push –* Push the local branch to the web repository

*Pull –* Pull the web branch to the local repository

*Pull Request –* Request to merge a remote branch into another(typically main)

**Example Github Flow**

First step when adding code to your project is to create a new local branch. After creating a local branch you then need to checkout the new local branch as the currently checked out branch is “main”. After checking out the new local branch you then can add all the code you would like to add for this version of the project. Git will automatically remember all the new code you added and all the code you changed. When you are finished you need to stage your changes. Github Desktop will automatically stage your changes to a branch. Next you need to commit the staged changes to your local branch. Github Desktop allows you to do this through the “commit” button. After committing you can then publish your branch on Github to create the branch on the remote repository and then push to add your changes to the remote repository branch. You can then create a Pull Request which will merge the remote repository branch with the remote repository “main” branch. Finally you can pull the changes so that your local “main” repository has the newly updated remote “main” repository changes.

Links:

https://github.com/