# Lab 2: Github and File Transfer

UWYO COSC 2030

## 1 Introduction to Github, Git, File Transfer, and Raspberry Pi

Github is used for version control. Github maintains a running record of your project, enabling you to revert changes, work across several systems and/or devices, and collaborate. Below are some terminal commands you will need to interact with Github:

- git clone [url] -initialize a Github repository on your machine
- `git pull` - checks the Github repository and updates your local files to reflect any changes
- `git add [file]` - adds the specified file to a commit `git add .` - adds all files in a repository to the commit
- `git commit -m ‘‘example’’` - creates a commit of all file changes with the message specified in quotes
- git push - pushes the commit to the remote (online) Github repository, changing the files on the remote repository to reflect the changes on your local machine

## 2 Authentication Tokens

For security reasons, you will not be able to access your Github account with your password from the command line. Instead, you will need to generate a passkey for your account and store it somewhere secure.

## 3 README.md Reminder

### 3.1 Please keep in mind that every lab must be submitted with a readme file in the following format. Points will be deducted for labs without readme files or incomplete readme files.

```
Your name
Help given/received
```

## 4 Lab 2 Instructions

### 4.1 Github GUI

1. S Follow this link to begin the assigment on Github Classroom: `https://classroom.github.com/a/s0TVyCFu`
2. Navigate to your repository and click through “Add File” -¿ “Upload existing file”
3. Upload a file of your choice (a .cpp file, an image, the .pdf for this lab, etc.)
4. Add a description of your file to the commit message
5. Click “Commit changes”
6. Check to ensure your files are uploaded

Edited by Milan Wolff, Michael Stoll, and Justin Wang for UWYO COSC 2030

### 4.2 Generating an Authentication Token

1. Navigate to your user account settings
2. At the bottom of the left sidebar, click "Developer Settings"
3. Select the option for a classic personal authentication token
4. Add a description to the note for the authentication token
5. Set the expiration date to the end of the semester
6. Select all boxes
7. Generate your passkey code
8. Save this code in a secure location
9. Use this authentication code as your Github password in later steps.

### 4.3 Git Command Line

- `ssh` into the Hive
- Copy the link from your Github repository
- Clone your repository, entering your username and authentication token as your password when prompted
- Use `ls` to verify that you successfully cloned the repository and use `cd` to navigate into the repository directory
- Using a text editor of your choice, make a basic Hello World .cpp program within this directory
- Using the commands above, add, commit, and push this file back up to Github
- Check Github to ensure that your file was pushed correctly

## 5 Using the Raspberry Pis

Follow the steps below to reserve and log into a Raspberry Pi

- Navigate to the course webpage
- Click on the "Pi's" link on the left sidebar. Click the "Claim" button
- ssh into the selected Pi by using the listed name as the host name, "padawan" as the username and "wyoming" as the password. Alternatively, you can use PuTTy. Example: Access Archer by entering `ssh padawan@archer.cs.uwyo.edu`
- Clone your Github repo here, and then navigate into the folder for your repository
- Run your Hello World program

Once you have successfully run your program, the lab is complete.

## 6 Submission

There are no deliverables for this lab.

Edited by Milan Wolff, Michael Stoll, and Justin Wang for UWYO COSC 2030