

This lab (Lab #2) is about Git (local repository), Github and a few of the commands you need to know to effectively manage an open source project.

Since we are already practicing git in class as part of our git lecture, you should not need the whole period. Accordingly, we will take the first few minutes to finish our discussion of git first.

You will be maintaining a repository on github for all your lab work. If you need help getting started, please check out the instructions for Lab 1. For this week you will need to create and commit a report labs/lab-02/report.md IN THE SAME REPOSITORY you used last week. You must maintain this lab as an open repository or Submittity will not be able to access your submissions. When you submit your lab to Submittity you will need to supply your github user name and the name of your fork. For me I would enter wdturner and oss-repo-template. Make sure that you commit your work and push it to your remote repository before you submit to Submittity!

The commands used in this lab will include git add, git commit, git pull, git checkout, git branch, git push, git log, git status, git diff, git tag, git rebase and git merge

- Use the command line (or git shell) for this lab

The command line is *required* - your ubuntu installation or gitbash will help if you decide to experiment with the command line

Useful pages:

- A cheat sheet for git commands
- A very nice book about git
- A cheat sheet for Markdown

Learning Objectives

In this lab you will practice the git commands we introduced. At the end of the lab you will be able to view and contribute to git repositories both from the command line and from the GitHub website. You will know about branching and will be able to work within a group to do joint development and to resolve conflicts within repository files.

Part 1

Generate ssh keys if you have not done so.

Follow these instructions for generating and adding ssh keys to Github (make sure you select the correct operating system at the top)

1. Setup git

- Follow the instructions from slides 7 - 27 of <https://github.com/rcos/CSCI-4470-OpenSource/blob/master/Modules/02.Git/Git-Introduction-Part-I/index.html> To view the instructions, you will need to clone the repository and open index.html in the correct local directory. This is a little bit of a “Catch 22” but you should be able to handle it.
 - Create a local repository **lab2part1** using the command line
- Create a README.md file.
 - This file should include, as a bulleted list:
 - * your name
 - * your photograph (Add it to your repository and use markdown to display it). If you are uncomfortable with a public photo, please find some meme or bitmoji to represent you.
 - * your graduating year
 - * your project sub-area (the type of project that you plan to work on). This does not need to be your final choice, just your current thought process.
- 2. Create a new repository on github and push the local repository and README.md file you created. > (*hint: you should be doing something like this*):

```
git init
git add README.md
git commit
git remote add origin <repo url>
git push origin master
```

Provide a link to this repository in your lab report file.

3. Create a file **first.py** that prints “Hello World!”.
- Add, commit, and push the file to your github repository on the master branch with **git push origin master**.
4. Create a new branch called **mars** with **git branch mars**.
 - Check out the mars branch by doing **git checkout mars**.
 - Change **first.py** to print “Hello Mars!”.
 - Add, commit, and push this file to the mars branch using **git push origin mars**.
5. Repeat with a branch called **jupiter**.
6. Visualize the branching diagram.
 - Use **gitk** and **gitk --all**
 - Compare it with to **git log --graph --oneline --decorate**.
- ***Include a screenshot in your lab report.***

Part 2

1. You can create a derivative of a repository by forking.
 - Read about forking here
 - Fork <https://github.com/octocat/Spoon-Knife> repository by pressing the Fork button in the top right corner of the repository page.
Provide a link to your fork of this repository in your lab report.
2. Clone the fork you made using `git clone <fork url>`
3. Add a file `myprojectprogress.md`
 - Write about your thoughts on possible projects of interest and how you would like to organize it. Common repository? Blessed repository? Other?
 - Push it back to your forked Github repository.
4. Do the first four levels (Introduction Sequence) of Learn Git Branching.
 - *Include a screenshot in your lab report.*

Part 3

1. Fork the repository <https://github.com/wdturner/PullReq>
 - Clone your fork using the command line.
 - In the `Summer2022` directory, add a file named `<firstName lastName>.md`, add today's date on the first line.
 - Add, commit, and push your file back to your forked version. Then use the `github` interface to make a pull request.
 - Ask one of the helpful instructional staff to merge your pull request.
 - You can update your repository to reflect changes in the upstream repository using:

```
git remote add upstream https://github.com/wdturner/PullReq.git
git pull upstream master
```

Provide a link to your fork of this repository in your lab report.

2. Make sure you understand the command `git diff` from this example
 - Examine the contents of the diff file from the previous question (your pull request) using `git diff HEAD~1`

Provide a screenshot of the diff to this repository in your lab report.

3. Understand git tagging from this example and this one
 - Tag your repository with `1.0.0`

Provide a screenshot of the output from `git tag -l` in your lab report.

5. As a table: (one person per table)

- Fork the repository <https://github.com/wdturner/OSSProjectIdeas.git>
- Add each member of the table to the repository:

Click 'Settings' on the repository page

Click 'Collaborators'

Add the username of each person

- Your fork is now a **common repository**

Please add a link to this repository in your lab report.

6. Each member should clone the **table's** repository

- Locally, in the Summer2022 directory, each member should create a file called ProjectIdeas<table number>.md
- Each person writes their project ideas inside their own version of the file.
Your ideas are not binding; it is just to get you thinking and to start communicating with your colleagues
- Each person should push to your **common repository**; fix any merge conflicts.
- Once everyone has their changes successfully in the model file, each table submits a pull request to the upstream repository.
- Resolve any merge conflicts that occur along the way.

– You can update your repository to reflect changes in the upstream repository using

```
git remote add upstream https://github.com/wdturner/OSSProjectIdeas.git
git pull upstream master
```

- Git Introduction Part II might be useful for this (open index.html in the same way as the Introduction Part I slides)

When you have completed the lab, submit your lab to Submittity by specifying your repository.