CMPS 3390 Lab 1

Fall 2024

Part 1 - Setup

1. For this assignment you will create a very simple c++ project to learn the basics of version control.
2. In your home folder on odin run the following commands:

```
mkdir -p 3390/lab1; cd 3390/lab1
```

   This will recursively create a 3390 folder with a lab1 folder inside of it, then change to that directory.

3. Inside of your lab1 folder, run the following command:

```
touch app.cpp README.md .gitignore
```

   This will create three empty files with the names specified.

4. Include the following code in **app.cpp**:

```
#include <iostream>
using namespace std;

int main(){
    cout << "WELCOME TO MY APP!" << endl;
    return 0;
}
```

5. Include the following code in **README.md**:

```
# Sample App For Learning Version Control

This is a simple app to learn about the follow:
- Version Control
- Git
- Github
- Markdown
- Staging & Committing
- Branching & Merging
- Push, fetch, and pull
- Forking & Pull Requests
```

6. Include the following code in **.gitignore**:

```
#ignore executables
*.out
*.exe

#ignore hidden "dot" files
.*

#whitelist specific "dot" files
!.gitignore
```

Part 2 - Git Basics

1. From your lab1 folder run the following command to initialize a local git repository:

```
git init -b main
```

   The "-b main" is to ensure that your initial branch is called "main" since github no longer supports "master".

2. To see the current state of your project type the following:

```
git status
```

   You will now see that you have a new git project/repo, but nothing has been staged or committed.

3. To stage all current files/changes and prepare them to be committed type the following:

```
git add -A
```

   If you check the status again you will see that the files are now staged.

4. To commit all staged changes with a brief commit message type the following:

```
git commit -m "Initial commit"
```

   Once again, check the status and you will see "nothing to commit, working clean tree".

   If you require a longer commit message, omit the -m and message. VIM will be opened instead.

5. Now go ahead and experiment with making changes to files, checking the status, staging files, and committing changes.

6. Once you have made a few more commits type the following command to see your commits:

```
git log
```

7. Create a branch from your main codebase and check it out using the following command:

```
git checkout -b math
```

8. Add a couple math functions to your app.cpp file, stage the changes, and commit them.
   Then, to merge those changes back into the main branch do the following:

```
git checkout main
git merge math -no-ff
```

Part 3 - Github Setup

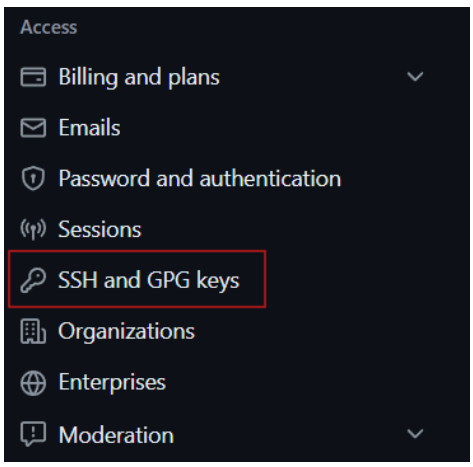
1. We will be using ssh authentication for our remote repositories, so **if you don't have a public/private keypair in your .ssh folder on odin**, run the following command:

```
ssh-keygen -t ed25519
```

   Hit enter a few times to accept the default options.

2. If you haven't already, go to https://github.com/ and create an account.

3. From your account settings, go to "SSH and GPG Keys":



4. Type the following command in odin:

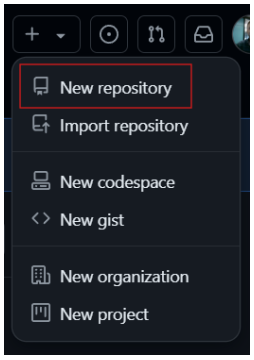
```
cat ~/.ssh/id_rsa.pub
```

   This will echo the entire contents of your public SSH key in the terminal. Copy it.

5. On the SSH and GPG keys page, click "New SSH Key", give it a title like "CSUB Odin", and paste the key.
   Now you will never have to login when you clone, push, or pull a repo from Odin.
   Notice you can add multiple SSH keys to your account, like from your own personal computer/laptop.

Part 4 - Creating a Remote Repository

1. From The plus dropdown on github, select "New Repository":



2. Give the repository a name, make it private, and click "Create Repository".
3. Once you have your repository create, it will give you an ssh URL for your repo something like this:

```
git@github.com:wmpaulroyer/CMPS3390-lab1.git
```

   Except it will have your username and repository name.

4. Finally, run the following commands from your lab1 folder on odin:

```
git config --global user.name "Your Name"
git config --global user.email "Your Email Address"
git remote add origin git@github.com:wmpaulroyer/CMPS3390-lab1.git
git push -u origin main
```

   But be sure to replace the sections highlighted in red with YOUR INFORMATION

CONGRATULATIONS! YOU HAVE SET UP YOUR GITHUB! BUT WE'RE JUST GETTING STARTED!

Part 5 - Advanced Topics

Time permitting during lab & lectures we will continue to go over advanced git/github topics including:

- Fething, Pulling, and Pushing Changes
- Adding Collaborators to your project
- Resolving Merge Conflicts
- Forking & Pull Requests
- Visualizing your git history
- Git tooling & integration into your IDEs
- THE POWER OF BLAMING OTHERS!