Labwork: Github

- If you are using windows computer in the lab, download Git portable: https://github.com/git-for-windows/git/releases/tag/v2.24.1.windows.2
- You will use the Git command line.
- From Git command line, read about some popular Git commands: *add, checkout, commit, diff, grep, log, rm, and status.*
- Use the following command to read about Git commands: *$ git help add*

GitHub setup

- GitHub is a free Git repository.
- Go to github.com and create an account for yourself.

Create a new GitHub account

- From the command line, configure Git with your user name and email address:

$ git config --global user.name "Joe F. Student"
$ git config --global user.email "joestudent@trentu.ca"

- use *git config --list* to display the contents of you config, and screen capture the part that shows you name and email are set correctly (you may need to use the down arrow to get the bottom of the file, and *:q* to exit it)

Labwork: Configuring Git and Forking a Project

- In Git bash (or terminal), navigate to your desktop (or a place on you filesystem you'd like to store your files).
- Clone the following repository: https://github.com/omalam/COIS2240.git
  $ git clone https://github.com/omalam/COIS2240.git
  - Cloning copies a remote repository into your local computer.
- Check the content of COIS2240 directory in your file browser. Take a screenshot
- Open Hello.pdf and take a screenshot

Pulling

- Now pull from the remote repository: *$ git pull*
- *Question#2: Copy the text that is on the file HelloAgain.txt*
- What happened is that I added another file to the repository. The pull command updates your local repository with the added files in the remote repository.

Labwork: Staging a File

- Now go to Github and create a **private** repository in your own account called *2240Lab2* and clone it in a directory in your machine.
- Create a text file called FirstName_LastName_StudentNo.txt inside the working directory for the repository on your machine. *(your first name underscore last name underscore your student number)*
  - Take a screenshot
- Now in bash, add this file to the ***staging*** area using the following command (remember staging from the class notes?):

  git add FirstName_LastName_StudentNo.txt
- Then run *git status* to view the current status and take a screenshot showing your staged file

Lab 2 - Committing a File

- Now commit this file to your local repository

```
git commit -m "committed my first file to Git."
```

-m is to write a commit message.

Take a screenshot of the results

Labwork: Pushing

- Now, push the committed file to the remote repository, and take a screenshot of the resulting messages

  git push origin

- origin is the default name that git gives to your remote repository. To find the full path for your remote repository, run the following command:

  git origin -v

- View your remote project in GitHub and see that your file is there (take a screen shot)

Labwork:Commit Logs

- You can view the commit logs by running the following command:
- *git log*
- Take a screenshot of your commit logs

- As a Git user, you should be familiar with the following three steps:

$ git add *your_file*
$ git commit -m*"your message"*
$ git push origin

Labwork: Checkout

- Now delete the file FirstName_LastName_StudentNo.txt from your directory, and check the status of the working directory using *git status* and take a screen shot
- Restore the deleted file from remote:
  - *$git checkout FirstName_LastName_StudentNo.txt*
- Take one last screenshot to show your returned file

Labwork: Configuring Git and Forking a Project

- How would you delete a file from a remote repository?
  - git rm *FirstName_LastName_StudentNo.txt*
  - git commit -m”removed the file”
  - git push origin

Showing Your Work

- Screenshots should include:
  - git config showing set name, email
  - contents of the COIS2240 folder,
  - an open Hello.pdf
  - the 2240Lab2 repository folder cloned on you machine with your text file in it
  - the status of your staged file
  - the results of the commit
  - you text file in your GitHub repository online
  - your commit logs
  - the status after delete
  - the directory containing your returned file