160 intro to git and make

- this isn't testable, just meant to provide overview/background on the makefiles and git commands we've been using
- make is a program to automate tasks, mostly focused on automating program compilation
- *for 160, I've provided makefiles to compile lab exercises and automate getting/submitting labs/projects*
- git is a version control system, allowing us to keep track of file modifications over time
- *the csci department has an added set of software tools so we can distribute/collect labs and feedback using git*

make, makefiles

- the make program is used to automate commands, usually for program compilation
- it looks for a “makefile” in the current directory, this makefile must contain instructions for the specific commands to use in that directory
- each command has
  - a target (the thing we want to make)
  - a list of files it uses (called the dependencies)
  - the sequence of commands to build the target
- to run the commands from in the directory, the user types

```
make thedesiredtarget
```

Our 160 makefiles: make160

- the makefile we put in our home directories (make160) has instructions on how to obtain and create the directories for each of our labs and projects
- if you look inside it (e.g. using pico) you'll see items like:

```
csci160/labex1:
   mkdir -p csci160/labex1
   ssh cscifork csci160-01/labex1 csci160-01/$$USER/labex1
   git clone csci:csci160-01/$$USER/labex2 csci160/labex2
```

- this is the sequence of linux/git commands it runs in order to build your csci160/labex1 directory from the instructor version

warning: don't alter the file content, you run the risk of breaking it

160 makefiles: in the labexes

- we also have makefiles (this time called makefile) in each of our lab and project directories
- these include the commands to compile the code, e.g.

```
labex2: labex2.cpp
    g++ -Wall -Wextra labex2.cpp -o labex2
```

- these also include the commands to submit the lab, e.g.

```
submit:
    git add -u .
    git commit -m “lab2 submit”
    git push --all origin
```

git

- git tracks of collections of files in a directory (repository)
- it allows the programmer to remember the state of all the files at different points in development (commit points)
- programmer can
  - specify file changes to remember (“git add” commands)
  - create checkpoints (“git commit” commands)
  - make copy a repository (“git clone” commands)
  - check for changes in the one we copied (“git pull” commands)
  - send our local changes back to the original (“git push” commands)

more information on git, make

- summary of the git lab processes for later csci courses:
  - csci.viu.ca/~wesselsd/guides/gitstudent.html
  - www.youtube.com/watch?v=6e0naRiyrOc
- csci 265 intro to make/makefiles
  - csci.viu.ca/~wesselsd/courses/csci265/slides/makefiles.pdf
  - www.youtube.com/watch?v=Q_hPztwLafc