CS221 - Lab02

Paul Haskell

INTRODUCTION:

In this lab you will install a few more software tools and will write a few simple C programs. After you're done testing, you will push your programs to GitHub. Of course you can push intermediate work to GitHub, so you don't lose it by accident.

Software Installation

For starters, please make sure you are running the "bash" shell:

- On **Windows**, run the "Git Bash" shell
- On **Mac**, just run 'bash' in a terminal window. (Later I will tell you how to have the terminal window run 'bash' automatically)
- And everyone, please copy the "**.profile**" file from the "Day01" directory in the CourseInfo repo to your "home directory". In bash, you can refer to your home directory as '**~**' (just the tilde character). For example:
  cd ~
  moves you to your home directory.
- I updated the **.profile** file late Monday evening in GitHub, so please fetch the latest.
- After copying, please edit the **.profile** file in your home directory. Near the bottom, there are some lines just for Windows and some just for Mac: please keep the appropriate ones and delete the others.

In CS221 we will use the "Gnu" compiler and tools.

- Mac users can use Homebrew to install the tools: from your Bash terminal, simply enter
  brew install gcc
- Windows users will have more work:
  - Browse to https://sourceforge.net/projects/mingw/files/Installer , download the installer, and run it
  - In the GUI, select the "base" and "g++" options. The go to the "Installation" menu and select "Apply Changes"
  - When this is done, it should have installed useful software in the /c/MinGW/bin directory on your computer
  - Finally, in your bash terminal, please run: **source ~/.profile**

Once you installed the tools, please check to be sure the installation was successful. In your Bash terminal, enter

```
gcc --version
```

(Note there are two hyphens back-to-back.) Did you get some version number, or an error?

Also try running

```
make --version
```

We will talk more about make during the next lecture.

Programs for this Lab

Your first program should be called **gituser.c** . Inside the `main()` method of this program, create a string variable whose value is your GitHub username. Please get the name exactly right, including capitalization. Add a second method to your program that takes a string as input and that returns an `int`. The method should take in a string, add up the ASCII values for each character in the string, and return the resulting integer. Your `main()` method should call this method with your GitHubId string and should print out the resulting integer, with no other text. That's it! The instructor will use this program to make sure the GitHub userid in his records for each student matches what each student thinks it is.

The second program is only a bit trickier. In a program called **checkers.c** , please print out the following pattern:

```
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
```

This sort of looks like a checkerboard, right? Please use some loops to print out this pattern—don't just have eight long printf statements.

CONCLUSION:

Please push both programs to GitHub before the lab deadline. The deadline for completion of Lab02 is 11:59pm Thursday January 30.

Hopefully you successfully wrote, compiled, and ran your first C program! The programs for this course will get longer and trickier in later weeks, but today should be a good start, making sure you can use all of the software tools we've talked about so far.

Future labs will have more involved scoring rubrics, and a portion of their scoring will be based on instructor-determined standards of Design Quality and Code Quality. For this lab, the scoring rubric is simpler:

Task	Score, points

Pushed **gituser.c** to your GitHub repo successfully	1
gituser.c compiles without errors	1
gituser.c prints the correct value	4
Pushed **checkers.c** to your GitHub repo successfully	1
checkers.c compiles without errors	1
checkers.c prints correct pattern	1
checkers.c prints correct pattern using loops, not 8 print statements	4