

# CSCI 1120 (Low-Level Computing), Spring 2020

## Homework 1

**Credit:** 5 points.

### 1 Reading

(None.)

### 2 Programming Problems

(For this assignment, you won't actually be programming, but you will be doing something on a computer, and submitting your answers in the way you'll submit your programs in later assignments.)

Do the following problem(s). You will end up with at least one text file and one PDF file. Submit your file(s) by sending mail to my TMail address with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 1120 hw 1" or "LL hw 1").

1. (5 points) (Not really a programming problem, but one that requires you to use a computer.) For this problem your mission is to learn a little more about traditional UNIX text editors `vi` and/or `emacs`. Do one or both of the following (full credit for doing one, extra credit if you do both).
  - Do something to improve your ability to use `vi` (or, more properly, `vim`, since that's what we have installed). Options include:
    - Start the interactive tutorial by opening a terminal window and typing `vimtutor`. Work through at least the first lesson, more if you have time.
    - If you've used `vi` before, think about your past use of `vi` and identify something you find particularly annoying (e.g., not knowing how to cut and paste). Then try to find a way to reduce the annoyance. You may find something helpful in the tutorial, or in the online help (which you start from within `vi` by typing `:help` and pressing the Enter key), or you may prefer to use your favorite search engine.
  - Learn something about `emacs`. If you've never used it, start it by typing `emacs -nw` in a terminal window. This should give you a page of instructions. Press control-h and then t to start an interactive tutorial. Work through as much of this tutorial as you need to in order to create and save a text file. (Starting the program by just typing `emacs` starts a graphical version of the program, which you may like, but the point here is to learn to use a text-mode editor, since in some situations that may be all you have access to.) If you already know something about `emacs`, either work through some of the tutorial, or do the second option for `vi` above (identify an annoyance and try to figure out a way to reduce it), but for `emacs`.

For each editor you learn about, produce two things:

- A text file, created with the editor itself, containing a paragraph or two about what you learned and what you wish you still knew about it.
- A “cheat sheet” you can use for reference when you use this editor. The point here is to have something you can keep on your desktop (virtual or physical) when you use the editor, until you remember the commands you find most useful. It could be a plain text file or a word-processor document or even a page of handwritten notes. Whatever you choose for it, send me either a plain-text or a PDF version of it.

Turn in the resulting files (plain-text “what I learned” file(s) and cheat sheet(s).)

### 3 Pledge

Include the Honor Code pledge or just the word “pledged”, plus *at least one of the following* about collaboration and help (as many as apply).<sup>1</sup> Text *in italics* is explanatory or something for you to fill in. For programming assignments, this should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please).

- This assignment is entirely my own work. (*Here, “entirely my own work” means that it’s your own work except for anything you got from the assignment itself — some programming assignments include “starter code”, for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the “sample programs page”.*)
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.* (*Here, “help” means significant help, beyond a little assistance with tools or compiler errors.*)
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.* (*Here too, you only need to mention significant help — you don’t need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.*)
- I provided help to *names of students* on this assignment. (*And here too, you only need to tell me about significant help.*)

### 4 Essay

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy. For programming assignments, it should go in the body of the e-mail or in a plain-text file `essay.txt` (no word-processor files please).

---

<sup>1</sup> Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM’s Special Interest Group on CS Education.