CSC 180F familiarization assignment, week of September 11, 2000

This ‘‘assignment’’ is just to familiarize you with the lab; it does not involve producing anything and it is not going to be graded. Experienced computer users can probably get through it very quickly.

The first official lab session will be this Monday, September 11. This is not the only time you can work in the labs, it’s just when they’re booked for our course and when a TA will be there. ***The lab session has been moved to SF1012 and SF1106.*** You still go at your assigned time (9:00-11:00 or 12:00-2:00). Due to scheduling constraints, in some cases there may be a CSC 180 TA in only one of the rooms, or only during the first hour.

I’d also like to take the opportunity to announce tutorial rooms. You have been assigned to a tutorial section, either Wednesday at 9:10 or Wednesday at 4:10. Please *ignore* the tutorial room assigned but we are *keeping* the assigned times.

The first tutorials will be on September 20. If you have been assigned to the W9 tutorial, then if your surname begins with a letter in A-G then please go to room GB304; if your surname begins with a letter in H-Z then please go to room WB219. If you have been assigned to the W4 tutorial, then if your surname begins with a letter in A-L then please go to room GB304; if your surname begins with a letter in M-Z then please go to room WB242.

---

1. Find a working, available computer and ‘‘log in’’. If you do not know your username (account name, user id, logname—all of these terms mean the same thing) and password, type ‘‘getname’’ for the username and just press return if it asks for a password. This will run a program which prompts you for some information and then reports your username and initial password. If you are in a ‘‘pc lab’’ (MC402, MC404, or WB316), you need to run a program called TeraTerm SSH to connect to the unix/linux computers.
(I use the term ‘‘unix’’ inclusively below, to include linux—there are many varieties of UNIX™, and the main reason linux isn’t one of them is the ™ part, not the UNIX part.)

2. Change your password by typing ‘‘passwd’’. Why does it prompt for your old password first? Why does it ask for your new password twice?

3. To create and modify your computer programs you will have to use an application program called an ‘‘editor’’. You can use any available editor you like, but a simple one to start with is called ‘‘pico’’. Type ‘‘pico hello.c’’ to begin editing a new file called ‘‘hello.c’’.

4. Here is a pared-down but valid C program. Type this in as the contents of hello.c, and save it, and exit pico. Pay attention to all of the punctuation symbols; they are crucial. For one, note that there is a difference between the ‘‘curly braces’’ (set brackets) and parentheses.

```
main()
{
    printf("Hello, world\n");
}
```

In fact there are no fewer than *three* minor things we would do differently in even this short program, but that’s the minimal version for the purpose of this familiarization session.

(over)

5. The ‘‘cat’’ command is the easiest way to display a short file on the screen in unix. Type ‘‘cat hello.c’’ to see your new file. To see a list of your files so far, type ‘‘ls’’.

6. To translate this C program into a machine language program so that you can run it, you need to run the C compiler, called ‘‘cc’’. Type ‘‘cc hello.c’’. If you do not get any error messages, you will now have a compiled version of this program called ‘‘a.out’’. Type ‘‘ls’’ to check.

7. Run your program by typing ‘‘./a.out’’. You may or may not be able to type simply ‘‘a.out’’, depending on how your account is set up. More about this soon. (MS-DOS users please note that that is a ‘‘forward slash’’. In fact MS-DOS is the only OS I know of which uses the ‘‘backslash’’ as a directory separator. Ask me some time if you want to hear a short rant about the history of this.)

8. You can also save the output into a file with a greater-than symbol; think of it as an arrow. Type ‘‘./a.out >x’’ to ‘‘redirect’’ the output to a file named ‘‘x’’. Type ‘‘ls’’ to see a list of all the files you have by now. Type ‘‘cat x’’ to see the contents of the file ‘‘x’’.

9. Type ‘‘mv x y’’ (for ‘‘move’’) to change the name of the file ‘‘x’’ to ‘‘y’’. Then try typing ‘‘ls’’, ‘‘cat y’’, and ‘‘cat x’’. Type ‘‘rm y’’ (for ‘‘remove’’) to delete the file. Type ‘‘ls’’. Also try the ‘‘cp’’ command (‘‘copy’’), which is like the ‘‘mv’’ command except it creates a new file, so that then there are two.

10. To log in to another computer over the network, use the command ‘‘ssh’’. There is a big server machine called ‘‘skule’’. Type ‘‘ssh skule’’ to log in there. Then type ‘‘./a.out’’ to run your program on skule. It doesn’t work! The problem is that skule is a different kind of computer than the one you’re sitting in front of, and each kind of computer has a completely different machine language. The C compiler on each machine produces machine language appropriate for that machine. So type ‘‘cc hello.c’’ *when you’re logged in to skule*, and then you can type ‘‘./a.out’’ successfully on skule (although no longer on the computer you’re sitting in front of!).

11. Log out of skule by typing ‘‘logout’’ (one word). You are still logged in to the computer you are sitting in front of.

12. As with editors, there are several possible e-mail reading programs you can use. The mail reader called ‘‘pine’’ is a good but simple one, and it has the same pico editor built in. Type ‘‘pine’’ to start it. Beware its habit of displaying important messages at the bottom. Your own e-mail address consists of your user name, an ‘at’ sign, and then ‘‘ecf.utoronto.ca’’. For example, if your user name is ‘‘betty’’, then your e-mail address is ‘‘betty@ecf.utoronto.ca’’. Send an e-mail message to a friend. Read an e-mail message sent by a friend.

13. There is a similar facility called ‘‘usenet news’’. With usenet news, the messages are not personal; all messages which anyone posts are available for all to read. You can read the ‘‘newsgroup’’ for this course by typing:

```
        rn -q ut.ecf.csc180
```

Do not post a test message. Wait until you have something to say, then post that.

The ut.ecf.csc180 newsgroup does not have any official status and we may or may not have time to read it or to respond to messages there. But it can be a useful resource for students to help each other, and we will read it some of the time.