CMSC6950 — Fall 2025
Homework #1

Due Date: Work must be submitted to GradeScope by 5pm (St. John's time) on Thursday, October 2.

The goal of this assignment is for you to practice with the basic shell commands discussed in the first classes. Each problem below asks you to explore one or a few such commands, and you should turn in either a screenshot of your terminal showing the commands and their output for each problem, or a pdf export of a Jupyter notebook with the terminal commands and output copy and pasted into Markdown cells.

1. (10 points) Open a Bash shell. Make a directory called `myname_CMSC6950` (where `myname` is your name), and change into that directory. Edit a text file in that directory, named `student_number.txt` that contains your student number, save the file, and show a directory listing that shows the date and time that you edited the file. Display the file's contents. Change back to your home directory.

2. (5 points) Open a Bash shell. Make an empty file called `copmutational_science.txt`. Fix the spelling mistake using Unix commands. Remove the file, remake the (misspelled) file, and fix the spelling mistake again using different Unix commands. You should complete all aspects of this question *only* using the command line (not using a text editor or other tools).

3. (5 points each) Download the file `alkanes.zip` from D2L and copy it to where you can find it in your Bash shell. Unzip the file (by running `unzip alkanes.zip`) and change into the directory it creates. Using a one-line command, answer each of the following questions:

   (a) What is the total number of lines in all of the files?

   (b) How many of the compounds have more than 6 atom's listed in their file?

   (c) List the files in order of the decreasing number of lines.

4. (5 points each) Using *regular expressions* enhances the power of the *grep* command. A regular expression is a string that includes characters (upper- or lower-case letters, numbers, spaces, punctuation, etc.) and one of the following aspects:

   - `[ABx]` to match only the characters "A", "B", and "x" in a position in the word
   - `.` to match any character, `\w` to match any letter or single digit
   - `^` to match the start of the line
   - `\s` to match a space, tab, or newline
   - `$` to match the end of the line
   - `\b` to match a word boundary
   - `*` to match the preceding character/element zero or more times
   - `+` to match the preceding character/element one or more times
   - `{n}` to match the preceding character/element exactly n times
   - `{n,m}` to match the preceding character/element between n and m times
   - `|` to mean or

- \ to match any of the above special characters

As examples,

- `grep -E "Fr[ea]nc[eh]" myfile.txt` will find all lines of `myfile.txt` that contain the words French and France, as well as the non-words of Frence and Franch
- `grep -E "^[Oo]rganize\b" myfile.txt` will find all lines of `myfile.txt` that start with the words Organize or organize and then a word boundary.

For this exercise, download the file `LittleWomen.txt` and use grep and regular expressions to find out how many lines of the file

(a) Begin with either the word "Meg" or the word "Jo"

(b) Begin with the word "The" followed by a space

(c) Contain a word with any letter, followed by qu followed by exactly six more letters

(d) Ends with either the word "time" or "times" followed by a period.