CSSSKL 142 - Lab 6

File Input/Output

University of Washington, Bothell

Summary

Today we will write some programs involving text file input and output. We will be importing some new libraries that contain methods necessary for making file i/o work.

Exercise 1 - File I/O

(*Statistics*) In a class called **Stats** (template is provided), given a text file of doubles (see this lab's Canvas page for a sample file), write a program that determines the average, the maximum, the minimum and how many numbers are in the following buckets: less than 0, between 0 (inclusive) and 100 (exclusive), and greater than or equal to 100.

Your program should print the following message to another file called "fileOut.txt":

Statistics for the numbers in fileIn.txt:

```
average: <average>
max: <max>
min: <min>
```

There are <negNum> negative numbers, <btw0and100> numbers between 0 (inclusive) and 100 (exclusive), and <geq100> numbers that are greater than or equal to 100.

(*Optional Challenge - work on it after completing the rest of the lab!*) Write a program that generates a text file that can be used as input for the problem described above. Use your knowledge of file output, loops and the random method from the Math library. How random and in what range the numbers should be it totally up to you!

Exercise 2 - More File I/O

In a class called **Diary** (template is provided), write a program that prompts users to enter the date as three integers separated by spaces (i.e mm dd yyyy) then it prompts them enter as many lines of prose they wish (for their to-do's list or diary entry). Your program should store their entries at the end of a file called "diaryLog.txt" and old data should NOT be erased. Each entry should be formatted as follows (example for a diary entry entered on 7/14/2017):

```
...old stuff.
Date: 07/14/2017
Dear diary, today we had a midterm. I think it went ok, ..
```

Exercise 3 – Advice Reading and Writing

Write a program (Advice.java) that gives and takes advice on program writing. The program starts by writing a piece of advice to the screen and asking the user to type in a different piece of advice. The program then ends.

The next person to run the program receives the advice given by the person who last ran the program.

The advice is kept in a text file `advice.txt` and the content of the file changes after each run of the program. Furthermore, you should have another text file called `adviceLog.txt`, which starts off blank but with each run of the program it logs the advice that was given (without deleting the previous advices). You should allow the user to type in advice of any length so that it can be any number of lines long. The user is told to end his or her advice by pressing the Return key two times. Your program can then test to see that it has reached the end of the input by checking to see when it reads an empty string "".

Input file: `advice.txt`

Don't Forget...

...to submit your files electronically to Canvas by due date!