COMP 3522 Lab #2: Let's dive in!

Christopher Thompson, Jeffrey Yim
jyim3@bcit.ca

Due Friday 11:59pm

Welcome!

Welcome back! In today's lab, you will use random numbers to populate a new file. Then you will use IO to read the file and perform a quick statistical analysis. Fun for all shall be had!

Requirements

Please complete the following:

1. Clone your repo using github classroom: https://classroom.github.com/a/NNIUJEnk
2. Fill out your **name** and **student number** at the top of **main.cpp**
3. Ensure you commit and push your work frequently. You will not earn full marks if you don't
4. Write a program that creates a file called Readings.txt. Inside the file, your program must create a list. The list is composed of integer double pairs. There is one pair per line. The integers are in sequence (0, 1, 2, 3, ...) beginning with zero and ending with some random value between **512 and 1024**. The doubles should be random values between **50.000 and 90.000**. The doubles only have **3 decimal places**. The file should look like this (of course your doubles will be random, and there will be more than 5 readings):

```
0 56.347
1 78.231
2 89.999
3 68.002
4 55.128
```

5. Write a program that opens the file called Readings.txt. The program must read the contents of the file and produce a report that looks like this to the screen:

```
There are XXX readings in the file. The average reading is YYY.YYY.
The highest reading is ZZZ.ZZZ.
The lowest reading is AAA.AAA.
The median reading is BBB.BBB.
```

6. Report breakdown:
   - `XXX` - number of int double pairs
   - `YYY.YYY` - average doubles value
   - `ZZZ.ZZZ` - highest doubles value
   - `AAA.AAA` - lowest doubles value
   - `BBB.BBB` - median doubles value

Output to console:

```
There are 574 readings in the file. The average reading is 70.245.
The highest reading is 89.998.
The lowest reading is 50.005.
The median reading is 70.660.
```

7. Restriction: Only use C style arrays to store your doubles

8. C++ mandatory reminder: **Don't put all your code in main.cpp**. Separate out the logic.
    - One example is to place the random number generation and file export code into. exporter.cpp/.hpp
    - then place the file input code and statistical analysis into, statistics.cpp/.hpp
    - finally, main will call code in each of the other files to tie the program together

9. Do NOT submit this Lab via D2L/The Learning Hub, submit it via GitHub

Grading

This lab will be marked out of 10. For full marks this week, you must:

10. (2 points) Commit and push to GitHub after each non-trivial change to your code
11. (3 points) Successfully write and test a program that creates a Readings.txt file exactly as described in the Requirements
12. (3 points) Successfully write and test a program that opens and consumes the Readings.txt file exactly as described in the Requirements
13. (2 points) Write code that is commented and formatted correctly using good variable names, efficient design choices, atomic functions, constants instead of magic numbers, thorough tests.

Hints

How do I generate text files in the same folder level as my other cpp and hpp files?
Prepend a **"../"** before the filename and it should place your generated text files in the same folder level as your cpp and hpp files

How do I achieve exactly three decimal places?

Getting exactly three decimal places for the doubles will require combining two of the output manipulators we learned about

How do I read an int then double?

The trick to reading an int, then a double from Readings.txt is similar to how we read an int then a double from the keyboard

```
int x, double y

cin >> x >> y; //the cin stream can read an int and then a double. Can we apply
similar logic/syntax to our file stream?
```

How do I read entire lines of text from a file?

The **getline** function, and **stringstreams** will be very helpful in this lab, check them out in the Week 2 lecture slides

How do I sort these doubles?

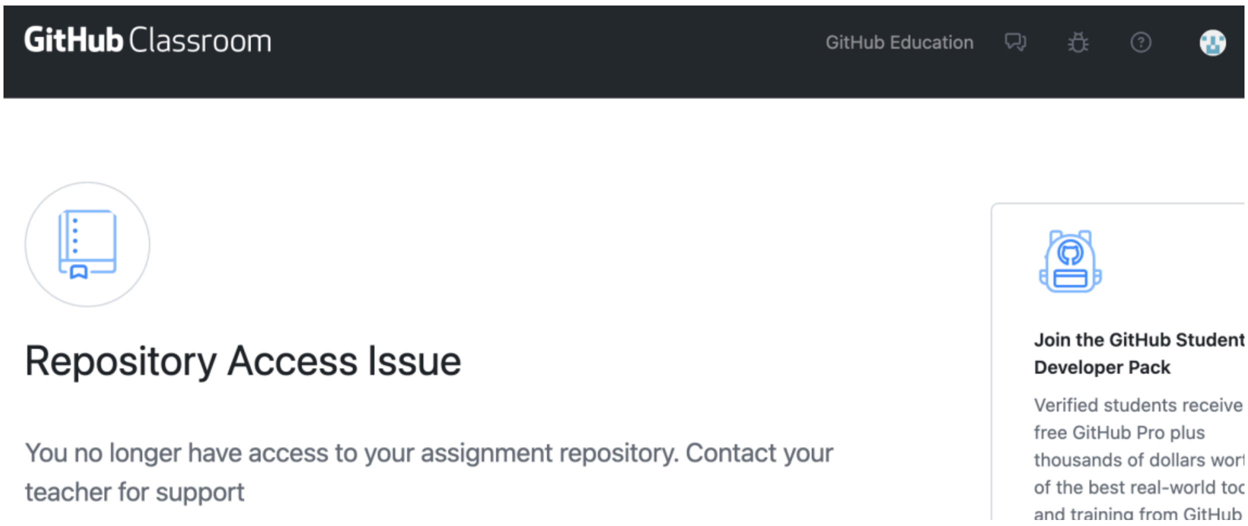
There's a built-in sort function that you can use

My filestream doesn't seem to work after I've read all the way to the end of file. How do I fix this?

That's right! When a filestream reads all the way to the end of a file, the end of file causes a failbit to be set in the filestream. Just call the `.clear()` function on your filestream and you should be able to use it regularly again.

Troubleshooting:

If you're getting an error cloning the Github class repo that looks like this:


GitHub Classroom
GitHub Education
Join the GitHub Student Developer Pack
Verified students receive free GitHub Pro plus thousands of dollars wort of the best real-world toc and training from GitHub

Repository Access Issue

You no longer have access to your assignment repository. Contact your teacher for support

Send a direct message on Discord to Jeff with your **github id**, and which **repositories** you are having issues with.

For example:
Github id: Jeff-bcit
Repos needed: lab 2 code

He'll message you back with an invitation link. Click on that link and accept it to get access