CSCI 1730 Breakout / Lab 02

Simple Programs

Last Updated: January 25, 2017

Problem / Exercise

Write C++ programs to perform the following tasks. In the program descriptions below, example input and output is provided. **NOTE:** You don't need arrays to solve any of these problems. You should NOT use arrays to solve any of these problems.

- `stat.cpp`: Let the user input a one or more integers, space separated, on a single line (as seen below), then work out and display the sum, average, sum of squares and population variance of the numbers. Remember, you do not need nor should you use an array to solve this problem.

```
Enter integers: 2 4 1 3
           Sum: 10.0
       Average: 2.5
Sum of Squares: 30.0
      Variance: 1.25
```

  NOTE: Given a set of numbers $x_1, x_2, \ldots, x_n$, the population variance be calculated by $\frac{1}{n} \sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right)^2$.

- `minmax.cpp`: Let the user input a one or more integers, space separated, on a single line (as seen below), then work out and display the largest and smallest values. Remember, you do not need nor should you use an array to solve this problem.

```
Enter integers: 1 9 3 7 5 6 4 8 2 10
           Min: 1
           Max: 10
```

- `numbers.cpp`: Write a program that prints out the binary, octal, decimal, and hexadecimal representation (respectively) of all unsigned shorts (in order). If the number is divisible by 3, then also print "Go". If the number is divisible by 5, then also print "Dawgs". Otherwise, "Sic'Em" should also be printed. Below is a sample of some of the entries near the middle:

```
0b1010'0000'0000'0000 0120000 40960 0xa000 Dawgs
0b1010'0000'0000'0001 0120001 40961 0xa001 Sic'Em
0b1010'0000'0000'0010 0120002 40962 0xa002 Go
0b1010'0000'0000'0011 0120003 40963 0xa003 Sic'Em
0b1010'0000'0000'0100 0120004 40964 0xa004 Sic'Em
0b1010'0000'0000'0101 0120005 40965 0xa005 GoDawgs
```

- `degrees.cpp`: Write a program to read a floating point number representing degrees Celsius, and print the equivalent temperature in degrees Fahrenheit. Print your results in a form such that the numbers are represented up to two decimal places.

```
Enter in a temperature in Celsius: 100
100.00 degrees Celsius converts to 212.00 degrees Fahrenheit.
```

- `time.cpp`: Given as input an integer number of seconds, print as output the equivalent time in hours, minutes and seconds.

```
Enter in the number of seconds as an integer: 7322
7322 seconds is equivalent to 2 hours 2 minutes 2 seconds.
```

Textbook & Other Resources

Students *may* find the following sections of the textbook useful for this assignment:

- DEITEL 2.1–2.8
- DEITEL 4.1–4.15
- DEITEL 5.1–5.14

Students *may* also find the following resources useful for this assignment:

- `std::cout` – http://en.cppreference.com/w/cpp/io/cout
- `std::cin` – http://en.cppreference.com/w/cpp/io/cin
- `std::dec, std::hex, std::oct` – http://en.cppreference.com/w/cpp/io/manip/hex
- `std::showbase` – http://en.cppreference.com/w/cpp/io/manip/showbase
- `std::getline` – http://en.cppreference.com/w/cpp/string/basic_string/getline
- `std::stringstream` – http://en.cppreference.com/w/cpp/io/basic_stringstream

1   Group Brainstorm

You are NOT allowed to use the computers during this time.

Breakup into groups based on your seating and brainstorm about how to solve the problem or exercise. Make sure everyone understands the problem, and sketch out potential ways to move towards a solution. Perhaps something that was discussed during lecture might be useful?

2   Submit Individual Brainstorm

You may use a computer from this point forward.

Login to eLC and submit a version of your group's brainstorm, written in your own words. You may add additional information if you want. You need to write enough in order to convince the grader that you understand the problem or exercise and that you have a plan for moving forward towards a solution. Please include the last names of the other people in your group in your submission. The brainstorm submission should be available on eLC in your assignment dropbox. You should submit your individual brainstorms before the end of your breakout period.

NOTE: Submissions that do not include an individual brainstorm will not be graded.

3   C++ Programs

3.1   Setup

Make sure that all of your files are in a directory called `LastName-FirstName-lab02`, where `LastName` and `FirstName` are replaced with your actual last and first names, respectively.

3.2   Code

For this lab, you should place your code into appropriately named files as described in the exercise description. Here is some more information about what is expected.

- All functions must be documented using Javadoc-style comments. Use inline documentation, as needed, to explain ambiguous or tricky parts of your code.

- The resulting executables must correspond to the source code filenames. For example, `stat.cpp` should result in an executable called `stat`. The expectation is that after your program is compiled and linked, the grader should be able to run each of your programs by typing something similar to the following:

```
$ ./stat
```

3.3   `Makefile` **File**

You need to make sure that you include a `Makefile`. Make sure that when you compile, you pass the following options to `g++` in addition to the `-c` option:

```
-Wall -std=c++14 -g -OO -pedantic-errors
```

3.4   `README` **File**

Make sure to include a `README` file that includes the following information presented in a reasonably formatted way:

- Your Name and 810/811#

- Instructions on how to compile and run your program.

- A Reflection Section. In a paragraph or two, compare and contrast what you actually did to complete the problem or exercise versus what you wrote in your initial brainstorm. How will this experience impact future planning/brainstorms?

Here is a partially filled out, example `README` file: `https://gist.github.com/mepcotterell/3ce865e3a151a3b49ec3`.

NOTE: Try to make sure that each line in your `README` file does not exceed 80 characters. Do not assume line-wrapping. Please manually insert line breaks if a lines exceeds 80 characters.

4   Submission

Before your next breakout lab session, you need to submit your code. You will still be submitting your project via nike. Make sure your work is on `nike.cs.uga.edu` in a directory called `LastName-FirstName-lab01`. From within the parent directory, execute the following command:

```
$ submit LastName-FirstName-lab02 cs1730a
```

It is also a good idea to email a copy to yourself. To do this, simply execute the following command, replacing the email address with your email address:

```
$ tar zcvf LastName-FirstName-lab02.tar.gz LastName-FirstName-lab02
$ mutt -s "lab01" -a LastName-FirstName-lab02.tar.gz -- your@email.com < /dev/null
```