

Breakout / Lab 02

Simple Programs

CSCI 1730 – Fall 2015

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 may use arrays if you want, but you don't need them.

- **stat.cpp:** Input a positive number of integers and work out the sum, average, sum of squares and population variance of the numbers.

```
Enter integers: 2 4
              Sum: 6.0
              Average: 3.0
Sum of Squares: 20.0
              Variance: 1.0
```

NOTE: Given a set of numbers x_1, x_2, \dots, 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:** Write a program that works out the largest and smallest values from a set of inputted integers. You may assume more than one integer is entered.

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

- **degrees.cpp:** Write a program to read a "float" representing a number of degrees Celsius, and print as a "float" 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.
```

- **endtime.cpp:** Write a program to read two integers with the following significance. The first integer value represents a time of day on a 24 hour clock, so that 1245 represents quarter to one mid-day, for example. The second integer represents a time duration in a similar way, so that 345 represents three hours and 45 minutes. This duration is to be added to the first time, and the result printed out in the same notation, in this case 1630 which is the time 3 hours and 45 minutes after 12.45.

```
Enter in a start time: 1415
Enter in a duration: 50
End time is: 1505
```

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. We prefer that you submit your individual brainstorms before the end of your breakout period, however, you generally have until 3PM on the day of your breakout (as indicated on eLC) to submit them.

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`.

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 line exceeds 80 characters.

4 Submission

Before the day of the next breakout 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
```