Assg 03: Process a File of Scientific Series of Data

CSci 515 Spring 2015

2015-01-30

Dates:

Due: Tuesday February 10, by Midnight

Objectives

- Be able to open a serial text file for reading and writing.
- Process more complex data items from a file of delimiter separated values.
- More practice with I/O stream formatting manipulators.
- Gain more practice using C control structures for implementing algorithms.
- Implement formula calculations into a typical data processing task written in C.

Description

In this assignment, we will be writing a filter, that will filter a file of delimiter separated values of data gathered from an experiment, perform some simple data analysis on the data, and save the results to a new file.

The input file you will be given has the following format (truncated, the real files will have many more items than 10 to process):

```
STRANGE
trial          x               y               z        class
00001    4.23169         4.68996        -8.86438        STRANGE
```

```
00002     5.43040        -3.59577        -4.71896        UP
00003     0.37792         3.34626        -0.22265        CHARM
00004     5.35208         3.96738         1.77813        UP
00005     0.90207        -0.38525         4.66088        CHARM
00006    -4.67474        -4.18064         1.65754        UP
00007     4.27666         4.56251        -8.56897        STRANGE
00008     4.17155         4.33587        -8.61905        STRANGE
00009     2.52960        -3.98237         8.60114        DOWN
00010     0.41158         5.61738         4.29349        UP
...
```

The first line represents a filter class upon which the data is to be filtered. More on this below.

The next line after the filter is a header for the columns/features of the data. After the header are the actual data trials in the experiment. Column 1 is simply the trial number for the row of data. Columns 2, 3 and 4 represent x, y and z measured positions of a physical component in some experiment. Column 5 represents a feature category, and will be a string.

Your task is to process this file. We are only interested in experimental trials that are of a particular class. In the above example, the first line indicates that we need to process the `STRANGE` trials. **NOTE**: I can and will use your code on different input files, that will filter on a different class, so you need to read in this first line from the file, and only process subsequent experimental trials of that type.

Each trial of the experiment has recorded the x, y and z position. We need to calculate the distance traveled between successive trials of the target filter class, using the standard euclidean distance:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)} \tag{1}$$

So for example, there are only 3 trials (1, 7 and 8) in our example data of the target `STRANGE` class. The distance traveled for the `STRANGE` item between trials 1 and 7 would be:

$$d = \sqrt{(4.27666 - 4.23169)^2 + (4.56251 - 4.68996)^2 + (-8.56897 - -8.86438)^2}$$

$$= 0.32486 \tag{2}$$

In addition to processing the file, you will open up a new file and save the results of the filtering and data processing to this file. The file I will give you to test with for input will be called `assg-03-data-in.dsv`. You should put the results into a file you open and write to called `assg-03-data-out.dsv`. Your results need to be formatted to look exactly like this output (using IOStream manipulators to format the output as required):

```
00007   0.32486
00008   0.25480
```

The first column is simply the trial identifier of the x_2, y_2, z_2 from the input (e.g. the second pair of coordinates used in the distance calculation), and the second column is the calculated distance. If you input file has N trials of the class being filtered for, then your output file will end up with $N-1$ distance values being calculated. The first column should be in a field width of 5, with leading 0's (as in the input). The second column is in a field width of 10, with 5 decimal places of precision and in fixed notation.

In summary you need to perform the following tasks for this assignment:

- Open the input file, and read in the trial class to filter on (and skip over the header).
- Open the input file and process it one line at a time.
- Open an output file, and output the formatted results of your distance calculations after filtering for successive trials.
- Calculate distance between successive trials of the trial class of interest.

Assignment Submission

An eCollege dropbox has been created for this assignment. You should upload your version of the out of class assignment by the end of Tuesday 2/10 (midnight) to the dropbox named `Assg 03 Time Series File`. Late submissions will not be graded.

Requirements and Grading Rubrics

Program Execution, Output and Functional Requirements

- Your program must compile, run and produce some sort of output to be graded. 0 if not satisfied.

- Your program must successfully open the file from the current working directory. 10 or more points.
- Your program must successfully open the output file for writing, and write some contents to it. 10 or more points.
- Your program must successfully read the lines from the file in the correct order and attempt to process them. 20 points.
- Your program must use I/O manipulators to achieve the correct output format to the output file. 10 or more points.
- Your program must correctly filter the input, and only calculate and output results for the trials of the correct class. 25 or more points
- Your program must produce the correct distance values between successive trials. 25 or more points.

Program Style

Your programs must conform to the style and formatting guidelines given for this course. The following is a list of the guidelines that are required for the lab to be submitted this week.

- The file header for the file with your name and program information and the function header for your main function must be present, and filled out correctly.
- You must indent your code correctly and have no embedded tabs in your source code. (Don't forget about the Visual Studio Format Selection command).
- You must not have any statements that are hacks in order to keep your terminal from closing when your program exits.
- You must have a single space before and after each binary operator.
- You must have a single blank line after the end of your declaration of variables at the top of a function, before the first code statement.
- You must have a single blank space after , and ; operators used as a separator in lists of variables, parameters or other control structures.

- You must have opening `{` and closing `}` for control statement blocks on their own line, indented correctly for the level of the control statement block.

Failure to conform to any of these formatting and programming practice guidelines for this lab will result in at least 1/3 of the points (33) for the assignment being removed. Failure to follow other class/textbook programming guidelines may result in a loss of points, especially for those programming practices given in our Deitel textbook that have been in our required reading so far.