

Simple Algebraic Calculations

One of the first things you will learn about C++ is how to perform numerical computations. In this project, you are given an incomplete program (see the end of this specification), which already includes the statements necessary to read input and specify output. You must add statements so that your completed program will perform the specified calculations.

The Problem:

Wind Chill

We've all heard weather reports that include the wind chill in the temperature. But what is it? Essentially, it seems fair to say that wind chill is the apparent, effective temperature as felt by bare skin in the presence of wind.

A couple of interesting articles and discussions are rooted at <http://www.nws.noaa.gov/os/windchill/index.shtml> and http://en.wikipedia.org/wiki/Wind_chill. In view of the cited resources, the currently-accepted formula for the wind chill adjusted temperature is:

$$T_{WC} = 35.74 + 0.6125T_F + (0.4275T_F - 35.75)V^{0.16}$$

where T_F is the air temperature in degrees Fahrenheit ($\leq 50^\circ$) and V is the wind speed in miles per hour (> 3).

Write a program to read an input file, described below, and use the formula given above to calculate the speed of the object. The program should then produce a report file, also described below.

Sample input and corresponding output:

Here is a sample input file, named `WCData.txt`, for the program.

Time	Air Temp	Wind Speed
02:15:00	54	12
02:30:00	52	5
02:45:00	47	21
03:00:00	48	17

The first two lines contain labels and will be ignored by the program. Each of the remaining lines will contain three data values, the time of the observation, the observed air temperature in degrees Fahrenheit, and the observed wind speed in miles per hour. The time will be given in the format HH:MM:SS, using a 24-hour representation. The air temperature will be an integer and the wind speed will be a non-negative integer. The data values on each line will be separated by white space characters.

Here is a sample output file, named `WindChillReport.txt`, for the program. The output file reports the results computed by the program. The output format must be specified rigidly because of the automated grading system (Curator) we will use in CS 1044.

Time	WC temp	WC Effect
02:15:00	50.0	-4.0
02:30:00	50.1	-1.9
02:45:00	39.0	-8.0
03:00:00	41.2	-6.8

The average adjusted temperature, based on 4 observations, was 45.1		

In this case, the output format would be specified as follows. The first line contains column labels for the reported data, and the second line delimits the labels from the data, and forms the beginning of a table. For this project, no points will be assigned to these two lines, so the precise text and formatting you use are unimportant.

There will be one line in the table for each line of report data in the input file. Each line of the table will include the time of the original observation, the computed wind chill adjusted temperature, and the difference between the adjusted and observed temperatures. The two numeric values will be reported showing one digit to the right of the decimal point.

The last data line in the table will be followed by a line to delimit the table, and a blank line. The final line of the output file will report the number of observations and the average wind chill adjusted temperature. In this case, as you already know if you have read the *Student Guide to the Curator*, it IS important that you use the same text as shown in the sample output file given above. The output should be aligned for easy readability.

Additional samples of input and correct output will be available on the course website.

Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to ten submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, can be found at:

<http://www.cs.vt.edu/curator/>

Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:
//
// - I have not discussed the C++ language code in my program with
//   anyone other than my instructor or the teaching assistants
//   assigned to this course.
//
// - I have not used C++ language code obtained from another student,
//   or any other unauthorized source, either modified or unmodified.
//
// - If any C++ language code or documentation used in my program
//   was obtained from another source, such as a text book or course
//   notes, that has been clearly noted with a proper citation in
//   the comments of my program.
//
// - I have not designed this program in such a way as to defeat or
//   interfere with the normal operation of the Curator System.
//
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.

The program source code:

```

// Project 1 for CS 1044, Summer I 2009
// Programmer:    <your name here>
// Last modified: <time/date of last change that you made>
//
// Purpose
// This program reads a sequence of temperature and wind speed observations,
// adjusts the reported temperature for wind chill, and produces a report
// file including the adjusted temperature and the amount of adjustment.
//
#include <iostream>    // for cout, if needed for debugging
#include <fstream>     // for file stream support
#include <iomanip>     // for formatting support
#include <cmath>       // for standard math functions, if needed
#include <string>      // for character string type
using namespace std;  // put all of that in scope

int main() {

    string obsTime;           // (read) time of observation
    int    obsTemp;          // (read) observed temperature
    int    obsSpeed;         // (read) observed wind speed
    double wcTemp;           // (computed) temperature adjusted for wind
    double wcTempSum = 0.0;  // (computed) sum of adjusted temperatures
    double minWCTemp = 150.0; // (computed) minimum adjusted temperature
    int    numObservations = 0; // (computed) number of observations

    ifstream Data("WCData.txt");           // open input file
    ofstream Log("WindChillReport.txt");    // open output file
    Log << fixed << showpoint;              // prepare output stream for decimal
                                           // output

    Log << "Time      WC temp      WC Effect" << endl; // write output header
    Log << "-----" << endl;

    Data.ignore(INT_MAX, '\n');            // skip over first two lines of input
    Data.ignore(INT_MAX, '\n');

    Data >> obsTime >> obsTemp >> obsSpeed; // read first line of observed data

    while ( Data ) {

        // Add C++ code to perform necessary computations on input data here:

        // Do not modify any of the rest of the code!

        Log << obsTime;                       // write data table
        Log << setw( 8) << setprecision(1) << wcTemp;
        Log << setw(14) << setprecision(1) << wcTemp - obsTemp;
        Log << endl;

        Data >> obsTime >> obsTemp >> obsSpeed; // try to read another line
                                                // of data
    }

    Log << "-----" << endl; // delimit end of table
    Log << endl;

```

```
// perform additional computations:

// write final line of report
Log << "The average adjusted temperature, based on " << numObservations
    << " observations, was " << setprecision(1) << avgWCTemp << endl;

Data.close(); // close the file streams
Log.close();
return 0;
}
```