Claude's Custom Counters, Inc. - *Bulk Data Input*

(Due Date: March 16; by 11:59 pm EST)

Overview

Your application proved to be a great success and resulted in Claude approving the next phase of development.

Sales volume has increased to the point that keyboard entry of order information is no longer effective. Now, regional sales representatives save their order information in text files that are consolidated at corporate headquarters. Our task is to develop an application that will read a sales file, validate entries, perform calculations, and generate summary output reports.

File Processing

Your software should prompt the user to enter the name and the path to the file containing sales data. Once that information is entered, your software must open the file and process the input data. If the input file fails to open, the software must output an error message indicating a problem occurred opening the file and a notice that processing cannot continue. Abnormal exits are not allowed, use if/else structure to ensure that code does not execute if the file fails to open.

If the file opens successfully, the software must read and process each line, or record, in the file. The first line of the file contains column headings. These are for anyone reading the file manually the program should read the entire line of headings to "get it out of the way" and then move on to the second line. The second line of the file is the first record of data that we will process. We will not know how many total records are in the file. We simply must continue reading and processing lines of data until reaching the end of the input file.   Each line of the file contains the following data elements:

Order Date	Date of order (string with format yyyy/mm/dd)
Delivery Date	Date of counter due to customer (string with format yyyy/mm/dd)
Stone Code	Stone code (a single character)
lenght	counter length (floating point number)
depth	counter depth (floating point number)
height	counter height (floating point number)
length edges finished	number of length edges polished/finished (an integer)
depth edges finished	number of depth edges polished/finished (an integer)
Order Number	Order Number (string, no spaces)
FIPS State Code	Federal Information Processing Standards State Code (string, no spaces)
Customer name & address	Customer's full name and address (string **with** spaces)

Data Validation

The system must test for a variety of possible data errors. You do not need to identify *data type* errors. That means that if a number is expected, then you may assume that the input data file will have a number. If a string is expected, you may assume the input data file will have a string, etc. The possibility of extraneous characters being present in the data file does not apply. Once you have extracted the value of one data element from the file, you may assume what follows is the value of the next data element.

As previously mentioned, if the input data file fails to open, then no calculations will be made, and no output will be sent to the terminal screen. Otherwise, for this version of the project, validation errors will **not** cause the processing to stop. If a data validation check fails, then no calculations will occur for ***that*** record. However, the values of that row of data will be output along with a list of the errors that were identified (see the sample output presented later). The table below lists Claude's business rules (some from Project #1, some new) that apply to data validation checks and other calculations. Data items not listed require no validation.

Data Item	Validation Rule(s)
Order Date	Will be in the format yyyy/mm/dd (see notes below)
Delivery Date	Will be in the format yyyy/mm/dd (see notes below) The delivery date cannot be the same as the Order Date, sales representatives occasionally make this mistake. The delivery date may be no more than 4 months after the order date (ignore the days, just count months), technicians sometimes build in some extra time for complex jobs, but Claude will not tolerate that.
Stone Code	Must be one of the valid character codes: M, m, G, g, Q, q
length	Minimum value 5.0, maximum value 25.0
depth	Minimum value 5.0, maximum value 25.0 (also must be less than or equal to length)
height	Must be between 58% and 80% of the depth
length edges finished	Can be 0, 1, or 2
depth edges finished	Can be 0, 1, or 2

IMPORTANT NOTES: When reading a date from the file, your software should separate the year, month, and day values and store them in integer variables; this applies to the Order Date and the Delivery Date. The year will always have four digits. With respect to the month, the format "mm" means two digits or one digit depending on the month. Similarly, the format "dd" could mean one digit or two digits.

Calculations and Output

As in Project #1, prices for materials offered are:

- Marble, at $92.99/sq. ft. installed
- Granite, at $78.99/sq. ft. installed
- Quartz, at $56.99/sq. ft. installed

The initial cost of a counter is based upon the area of material required for fabrication. When pieces are cut some material is wasted, so we add 26% to the area of the finished piece and then round any fractional value up.

Exposed edges can be finished by smoothing and polishing for $4.99 a linear foot.

Your software must calculate the amount of material required to begin fabrication, the cost of stone, the cost of edge finishing, and the total cost for the installed product.

Program Output

When the program runs, it shall output selected values from the input data file. If a record contains errors, also output error message(s). Keep running totals of the records processed, records with errors, and records without errors. For records without errors, also keep running totals of the counter tops by material including the number of counters of each type of stone, the total amount of material to begin fabrication, and the total cost. Output these running totals in a summary table after all records have been processed.

Sample output from an execution of the program:

```
  Order    Delivery   S                    L   D       Sq.      Total
   Date       Date    C   Len.   Dep.   Hei.   E   E        Ft.       Cost
--------------------------------------------------------------------------------
2015/01/20  2015/04/20  q   5.60   5.00   3.35   0   1      24.00    1392.71
2015/12/24  2016/02/24  Q  19.00  17.80  13.35   0   2     320.00   18414.44
2015/05/07  2015/08/07  Q  17.60  16.20   9.40   2   1     209.00   12167.40
2015/06/26  2015/07/26  q  11.40  11.40   7.52   0   2     109.00    6325.68
 ▪ ▪ ▪

2015/05/15  2015/08/15  m  11.10   6.80   4.96   2   1      70.00    6654.01
2015/04/06  2015/04/06  M  21.10   8.40   0.05  12  12
            ERROR: order date and delivery date are equal.
            ERROR: Invalid height.
            ERROR: invalid number of length edges to finish.
            ERROR: invalid number of depth edges to finish.

2015/11/02  2016/09/02  g   2.30   5.10  93.47   0   2
            ERROR: delivery date is too far from order date.
            ERROR: Invalid length.
            ERROR: Invalid depth.
            ERROR: Invalid height.
 ▪ ▪ ▪

2015/02/12  2015/03/12  M  24.00   5.30   3.66   2   2     111.00   10614.30
2015/02/18  2015/05/18  M  15.40   5.50   3.85   0   0      75.00    6974.25
2015/09/28  2015/10/28  Q   7.10   6.80   4.15   0   1      38.00    2199.55
2015/01/08  2015/03/08  g   9.80   5.00   2.70   0   0
            ERROR: Invalid height.

2015/10/18  2015/12/18  O  19.40   5.20   4.16   1   2
            ERROR: stone code is not a valid value.


Counts: Total Records = 12972   Records with Errors = 3911   Records without Errors = 9061

           TOTALS (records without errors)
   Stone    Count      Square Feet           Cost
-------------------------------------------------------------------
   Marble    3458        507976.00       47673578.71
  Granite    3402        486068.00       38813997.17
   Quartz    2201        306976.00       17771105.55
-------------------------------------------------------------------
```

Program Source Code

Important: Your output and input should be very similar to that shown in the sample output. Some content must also be included in your program **exactly** as specified.

Academic Integrity

This is an individual project and all work must be your own. Refer to the guidelines specified in the *Academic Honesty* section of this course syllabus or contact me if you have any questions.

Include the following comments at the start of your source code file:

```
/*
 * <FileName>.<file extension>
 *
 *  COSC 051
 *  Project #2
 *
 *  Author: <your name>
 *
 *  In accordance with the class policies and Georgetown's
 *  Honor Code, I certify that, with the exception of the
 *  class resources and those items noted below, I have neither
 *  given nor received any assistance on this project.
 *
 *  References not otherwise commented within the program source code.
 *  Note that you should not mention any help from the TAs, the professor,
 *  or any code taken from the class textbooks.
 */
```

These comments must appear **exactly** as shown above. The only difference will be values that you replace where there are "place holders" within angle brackets such as `<netID>` which should be replaced by your own `netID`.

Project Submission Details

You will upload your `.cpp` file containing your source code onto the Canvas under project-2. Do **not** post your executable file. You should ensure that your source file compiles on the server and that the executable file runs and produces the correct output. Use the following file name for your file: `<netID>P2.cpp.` See rubric for the point reductions for the late submissions.

Programming Skills

The programming skills required to complete this assignment include:

- Screen output (cout)
- Keyboard input (cin)
- Basic data validation
- Basic output formatting
- Basic calculations

- **File input/output**
- **Control structures for repetition**
- **Advanced output formatting**
- **Tabulated output**
- **Advanced data validation**

How to approach this program

For this project you are **NOT** submitting your design. However, as you learned from your first project, you should first design your software before start coding. Make sure that your code compiles and runs prior to moving forward from addressing one problem to the next.

Write the Pseudo code (design) of your program (this is to help you but you do NOT submit this) for grading:

- Study the project description carefully, begin thinking about your design, post your questions on the discussion board on Canvas
- Create a document to store your design
- Identify and add global constants
- Change prompts, just need to ask for full file path and filename (this is one thing, not two separate pieces)
- Add file open, test for file open success, and file close
- Add loop to process file
- Add data validation
- Add row-by-row calculations
- Add "tally" calculations (running totals)
- Add summary calculations
- Add output
- Make any modifications or other design refinements needed

Start Coding by creating milestones for yourself:

Milestone 1 (Code) –

- Create an empty source code file; insert heading comments, add preprocessor directives, add using namespace std;
- Add a "skeleton" of function main()
- Compile and run (nothing much should happen, this is just a check for compiler errors)
- Add global constants
- Add data input prompt statement and input statement for the path/filename of the data file
- Add variable declarations and initializations (some maybe coming from your 1st project)
- Compile and run after each addition

Milestone 2 (Code) -

- Add code to open the data file and test for success, if the file fails to open output an error message but do not attempt to read the file, if the file does open it is good to also output a "success" message
- Add a loop to read each line of the file, **this is just a test**, read the entire line into one string and then output that string to the terminal, once all records have been read, close the file

Milestone 3 (Code) –

- Now, add code to read each line, but extract the individual data fields in the correct variables (instead of one long string)
- Add data validation code
- Add detailed output code, includes values from each row read from the file, and error messages for any records that did not pass validation
- Add code to make calculations for records without errors

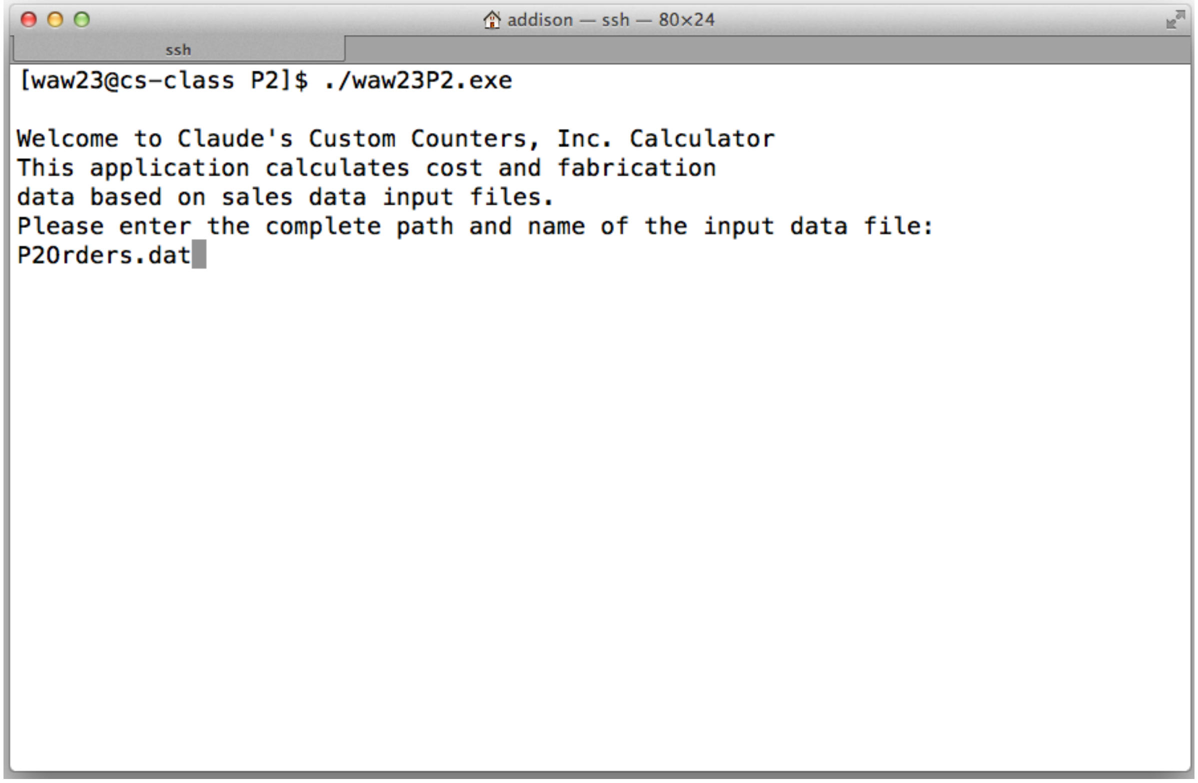
Milestone 4 (Code) –

- Add code to keep running totals and calculate summary statistics
- Add formatted summary output

Milestone 5 (Code) –

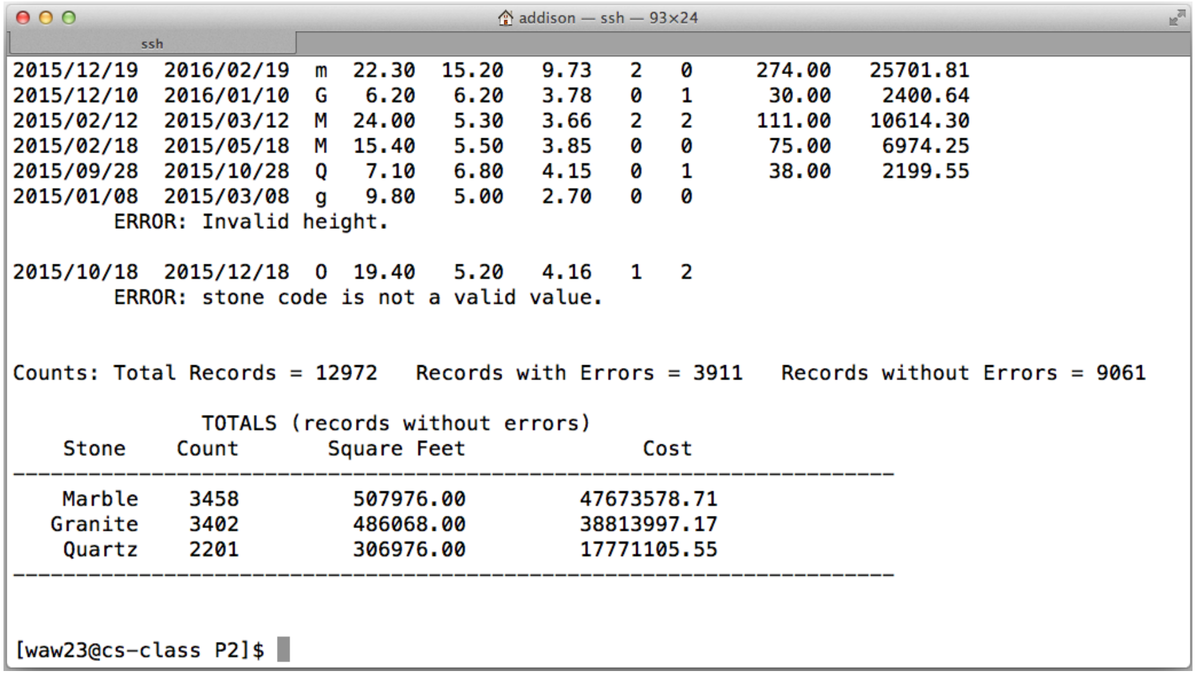
- Fine tune calculations and output
- Ensure program compiles and runs on the server
- Turn-in code… on time

Screen Captures



```
[waw23@cs-class P2]$ ./waw23P2.exe

Welcome to Claude's Custom Counters, Inc. Calculator
This application calculates cost and fabrication
data based on sales data input files.
Please enter the complete path and name of the input data file:
P2Orders.dat
```



```
2015/12/19  2016/02/19  m  22.30  15.20   9.73   2   0     274.00   25701.81
2015/12/10  2016/01/10  G   6.20   6.20   3.78   0   1      30.00    2400.64
2015/02/12  2015/03/12  M  24.00   5.30   3.66   2   2     111.00   10614.30
2015/02/18  2015/05/18  M  15.40   5.50   3.85   0   0      75.00    6974.25
2015/09/28  2015/10/28  Q   7.10   6.80   4.15   0   1      38.00    2199.55
2015/01/08  2015/03/08  g   9.80   5.00   2.70   0   0
        ERROR: Invalid height.

2015/10/18  2015/12/18  O  19.40   5.20   4.16   1   2
        ERROR: stone code is not a valid value.


Counts: Total Records = 12972   Records with Errors = 3911   Records without Errors = 9061

               TOTALS (records without errors)
    Stone    Count        Square Feet               Cost
----------------------------------------------------------------------
    Marble    3458         507976.00         47673578.71
   Granite    3402         486068.00         38813997.17
    Quartz    2201         306976.00         17771105.55
----------------------------------------------------------------------


[waw23@cs-class P2]$
```