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

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)
length	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.

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 = 3906   Records without Errors = 9066

           TOTALS (records without errors)
   Stone   Count      Square Feet          Cost
-------------------------------------------------------------------
   Marble   3459        508217.00        47696304.67
  Granite   3404        486490.00        38848129.85
   Quartz   2203        307157.00        17782092.39
-------------------------------------------------------------------
```

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 <term year>
 *  Project #2
 *
 *  Due on: <Due Date>
 *  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`. For example, I would replace `<netID>P2.cpp` with `waw23P2.cpp`.

Submission Details

Upload (as instructed by your professor) a `.cpp` file containing your source code. 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>P1.cpp.` Late submissions will be penalized heavily – see rubric for details. If you are late you may turn in the project to receive feedback but the grade may be zero. In general, requests for extensions will not be considered.

Programming Skills

The programming skills required to complete this assignment include:

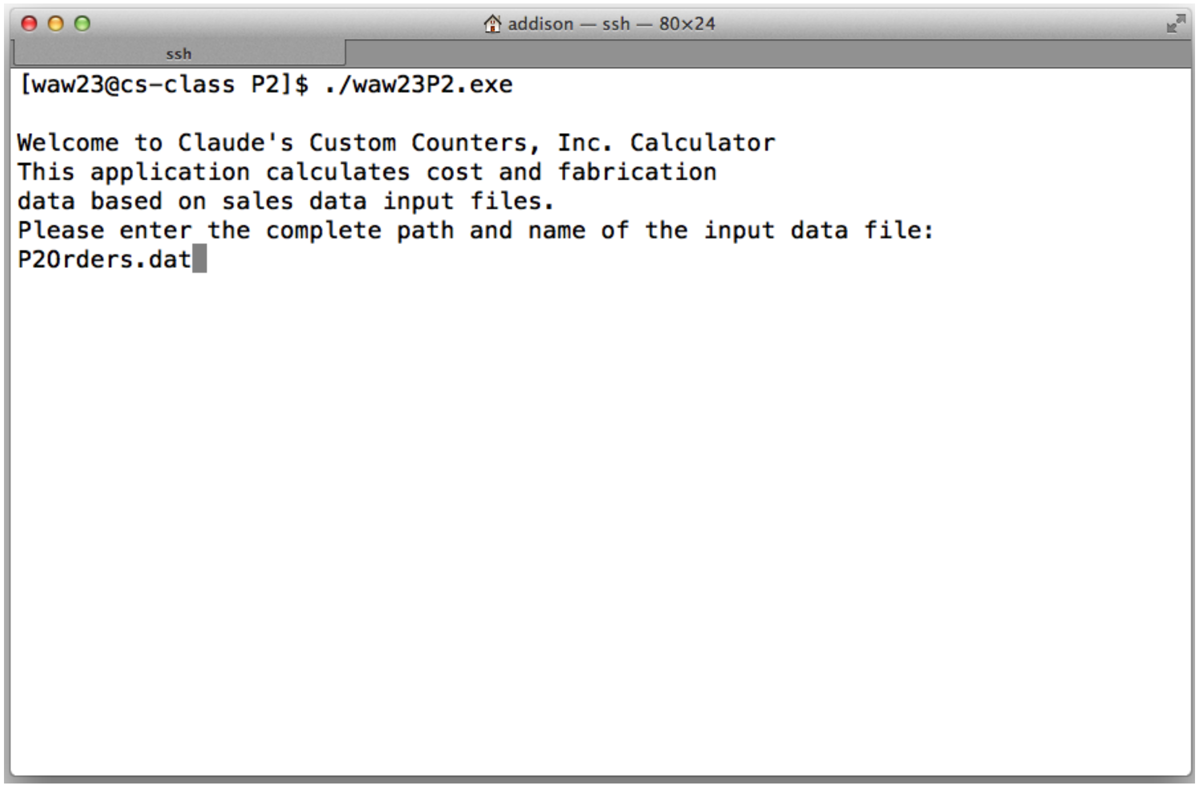
- Screen output (cout)
- Keyboard input (cin)
- Basic data validation
- Basic output formatting
- Basic calculations
- **File input/output**
- **Control structures for repetition**
- **Advanced output formatting**
- **Tabulated output**
- **Advanced data validation**

Grading

	Grade Standards - Missing: 0%, Poor: up to 50%, Fair: up to 67%, Good: up to 82%, Excellent: up to 99%, Perfect: 100%	
	Detailed Rubric (Code)	**100.00**
1	**Code Quality and Formatting**	***14.00***
	proper indentation	
	good variable and constant names	
	good use of constants (no "magic numbers" in calculations)	
	good use of comments	
	good use of vertical white space to separate code	
	good use of horizontal white space to improve readability	
	line length less than 100 characters	
2	**User interface / data input**	***17.00***
	outputs a brief greeting message	
	outputs prompts user correctly	
	repeated user prompts are in correct order	
	values entered by user are input into named variables of appropriate data type	
	any extraneous characters entered after a valid entry are ignored	
	error messages are clear and descriptive	
	for character input, both uppercase and lowercase are accepted as valid	
3	**Data validation algorithms**	***20.00***
	all input data are validated to ensure they are valid and/or within limits	
	prompts for data input are in reasonable order, test for errors and exit as soon as possible (don't make the user keep entering data if there has already been a fatal error)	
	if any input data fail validation error message(s) are displayed	
	processing terminates (gracefully) if any data fail validation, "abnormal" exits will be allowed (in moderation) for Project #1, but will eventually be prohibited	
4	**Calculation algorithms**	**37.00**
	geometric quantities are accurately calculated using the correct equations provided	
	numeric quantities are accurately calculated	
5	**Output**	**12.00**
	outputs are correct corresponding to the user's entries	
	outputs display correct calculations	
	output is neatly arranged on screen and is consistent with the output shown in the example program	

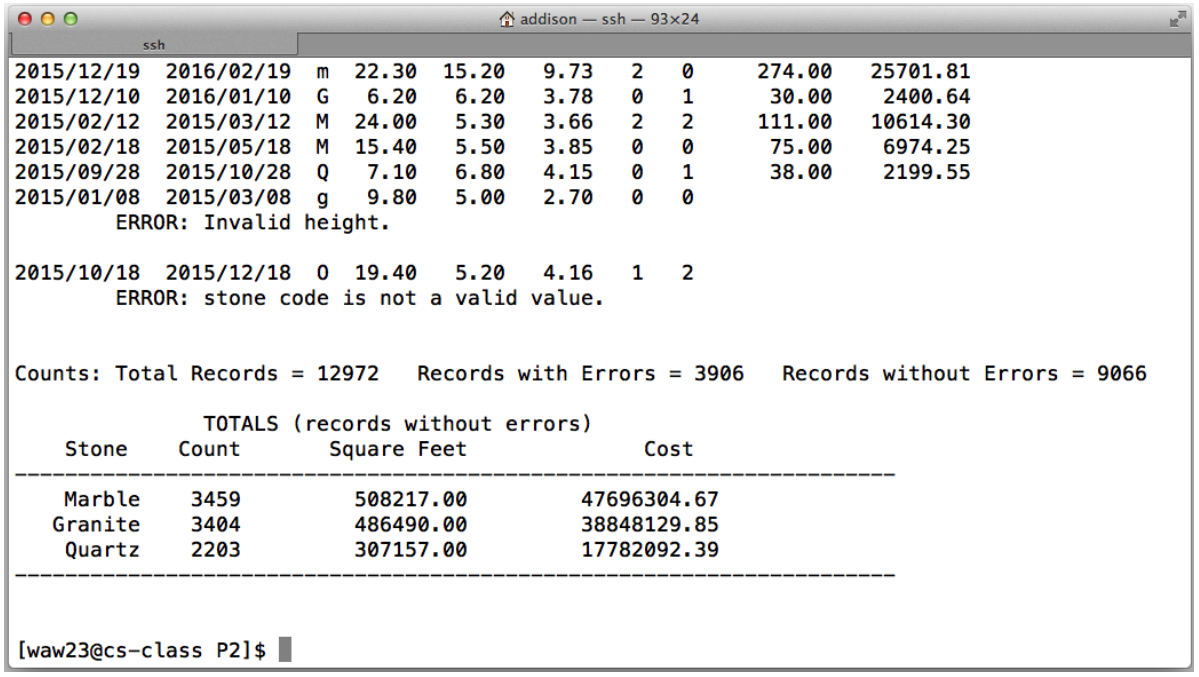
	Common Deductions (Code)	
	Program does not compile ON THE CLASS SERVER (deduction varies depending on how bad, value listed is max)	-100.00
	Program compiles but has warnings ON THE CLASS SERVER (deduction varies depending on how bad, value listed is max)	-100.00
	Program crashes during execution ON THE CLASS SERVER, or abnormal exit, or multiple exit points per method	-100.00
	Code uses any global variables	-40.00
	Filename does not follow conventions specified	-20.00
	Required comments and honor statement not included at start of file exactly as specified	-30.00
	Late penalty for each 15 minutes late	-2.50

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 = 3906   Records without Errors = 9066

               TOTALS (records without errors)
    Stone    Count       Square Feet              Cost
----------------------------------------------------------------------
    Marble    3459         508217.00          47696304.67
   Granite    3404         486490.00          38848129.85
    Quartz    2203         307157.00          17782092.39
----------------------------------------------------------------------


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

THIS PAGE INTENTIONALLY LEFT BLANK