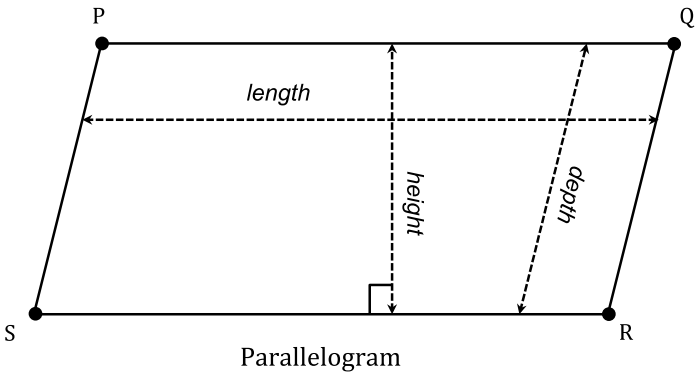
COSC 051 (See Canvas for Due Date info)

Claude's Custom Counters, Inc.

Background

Claude has built a booming business designing high-end custom counters for discerning customers. Claude's motto, *right angles are just wrong*, guides his every design. While most vendors deliver boring rectangular designs, Claude will only produce parallelogram counters having interior angels that are not 90˚. As a boutique designer for eccentric customers Claude was previously able to make all calculations in his head. However, demand for his custom designs recently went mainstream and he is now in serious need of automation. You must develop a computer program to determine how much one of Claude's custom countertops will cost.



Parallelogram

$$Area = length \times height$$

Each countertop piece is cut from stone blocks. Materials currently 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. Because of wastage when pieces are cut, we must add 26% to the area of the finished piece and then round that up to the nearest whole number.

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

Program Input

- The type of stone (first letter of the stone name)
- The length of the countertop
- The depth of the countertop
- The height of the countertop
- How many length edges and depth edges are to be finished

Each item of input requires validation. The minimum value for both length and depth is 5 feet. The maximum value for both length and depth is 25 feet. The length must be greater than or equal to the depth (if the length and depth are equal, the counter is a rhombus). The height must be between 58% and 80% of the shortest side:

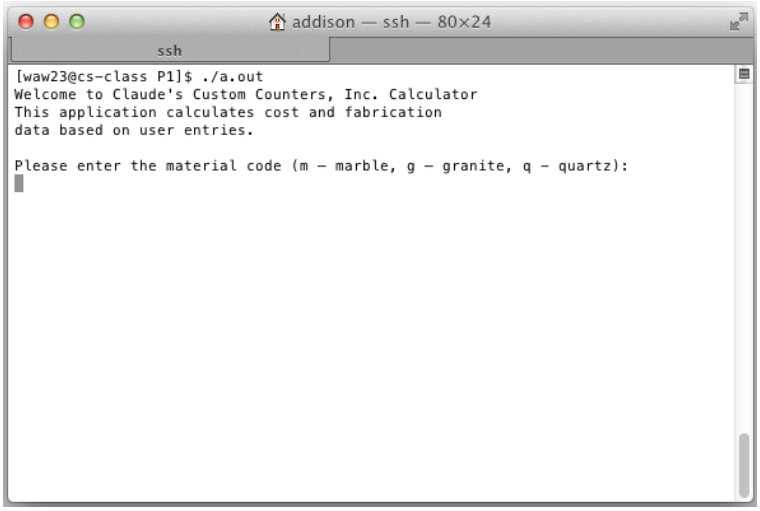
$$5.0 \leq length \leq 25.0$$
$$5.0 \leq depth \leq 25.0$$
$$depth \leq length$$
$$0.58\ (depth) \leq height \leq 0.80\ (\text{depth})$$

The type of stone must be one of the three options specified above. The number of finished length edges must be an integer between 0 and 2. The number of finished depth edges must be an integer between 0 and 2. After each value is entered, the value shall be tested to ensure that it is valid and/or within the appropriate numeric range. If any entry is invalid an error message shall be displayed and the program will end. You are not required to test for data type errors. This means that if a character value is expected you may assume that the user entered a character, if a number is expected, you may assume that the user entered a number. For character values (such as the first letter of stone names) your software shall accept both upper case and lower case letters as valid. Your software shall also disregard any extraneous information typed after a valid entry. For example the lower case letter g would be valid for granite, the upper case letter G would be valid for granite, as would the entire name Granite (all characters after the G are ignored). If all values entered are valid, then your software shall make the required calculations and output the results specified below.

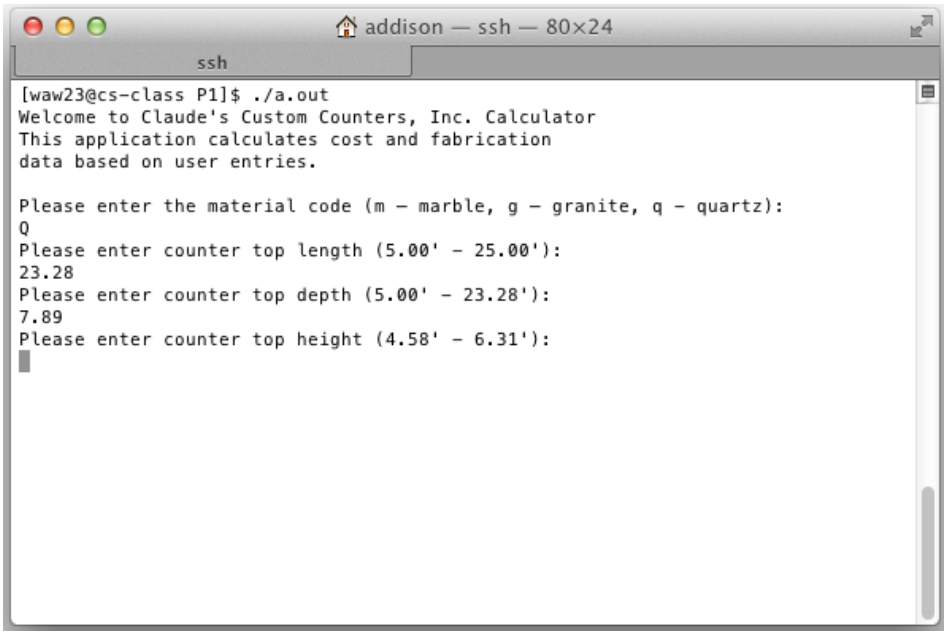
Program Output

- The length, depth, and height of the countertop
- The number of square feet of material needed to begin fabrication
- The cost of the stone
- The cost of the edge finishing
- The total cost of the above

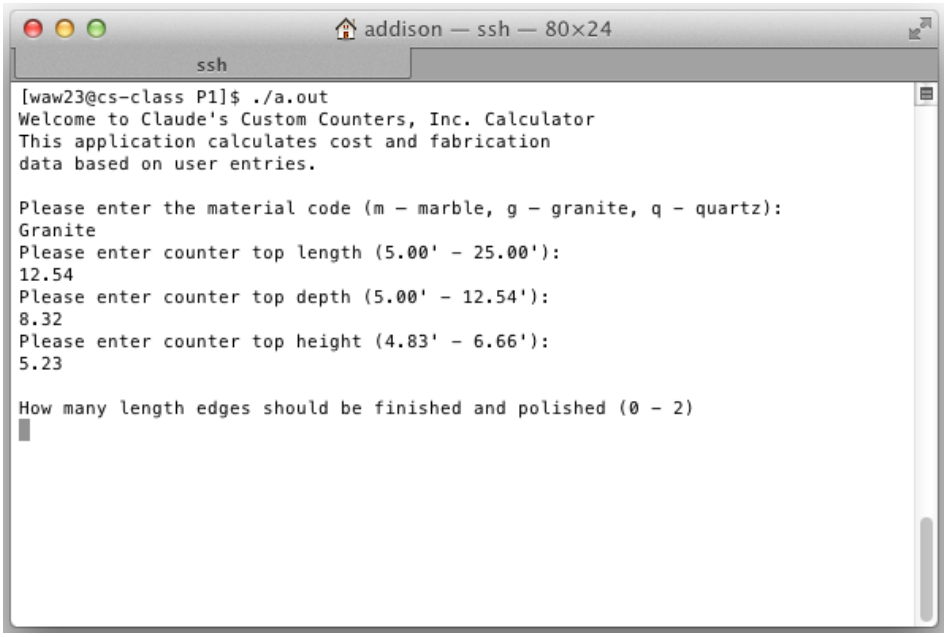
Screen Captures



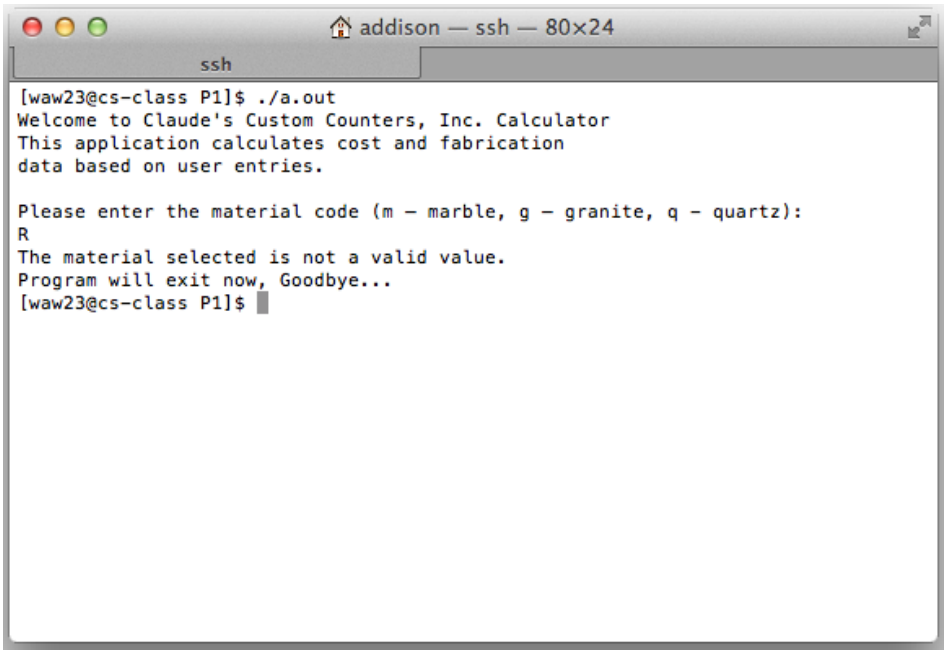
Your software shall have "smart quotes" that inform the user of acceptable entries:



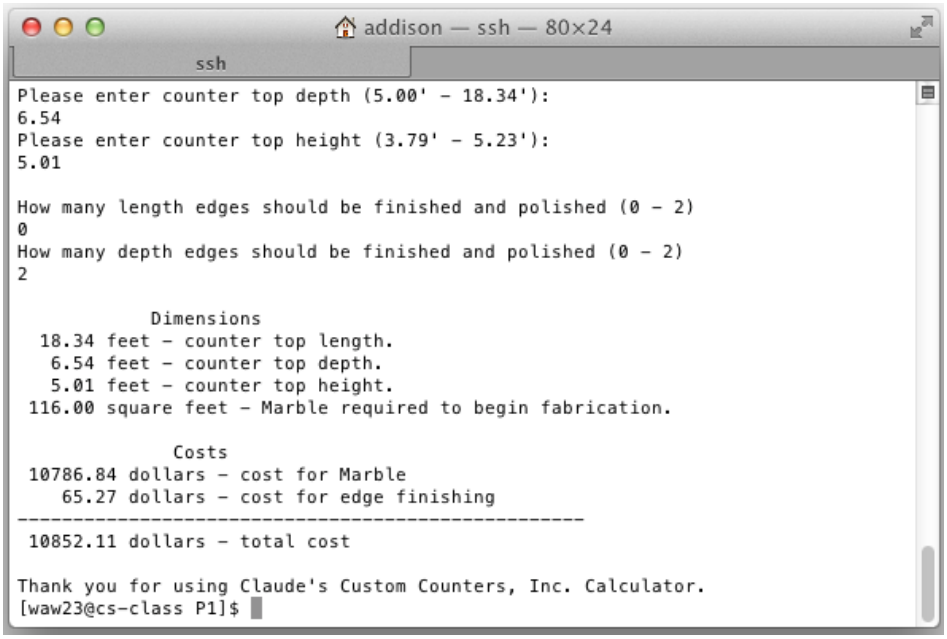
Your software shall accept upper case and lower case for character data entry. It shall also ignore any extraneous characters entered after a valid value is entered. Note in the screen capture below, the entry Granite is accepted. The upper case character G is stored as the user's entry. The remaining characters are ignored.



Your software shall display clear error messages for any invalid entries. If an invalid value is entered, the program should exit after displaying the error message.



Your software shall display neatly formatted, accurately calculated output.



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.

Part A - Design Document (Submit by the specified deadline)

Include the following comments at the start of your design document file:

```
/*
 * <FileName>.<file extension>
 *
 *  COSC 051-<put section# here> <put year and term here>
 *  Project #1
 *
 *  Due on: <put due date here>
 *  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.
 */
```

For the first part of this project you must submit a pseudocode design document showing the algorithm(s) you plan to implement. Our pseudocode language for the project contains the following terms:

START
INPUT
OUTPUT
CALCULATE
IF *condition*, **THEN** *statement*
IF *condition*, **THEN** *statement*; **OTHERWISE**, *statement*
STOP

If you need to group multiple statements together, say in an **IF** statement, use

BEGIN
statement
...
statement
END

Part A - Submission Details

Upload a .pdf file containing your design using the language described above. Use the following file name for your file: `<netID>P1.pdf` (replace `<netID>` with your netID and remove the angle brackets). 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 unless due to health issues or emergency reasons.

Part B - Program Source Code (Submit by the specified deadline)

Important: Your output and input should be very similar to that shown in the sample output. Please ask for the input in **exactly** the same order shown and only request the same items shown - do not ask for any other input. This will assist in grading your program. Some content must also be included in your program **exactly** as specified.

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

```
/*
* <FileName>.<file extension>
 *
 *  COSC 051-<put section# here> <put year and term here>
 *  Project #1
 *
 *  Due on: <put due date here>
 *  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`.

Part B - Submission Details

Upload 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 unless due to health issues or emergency reasons.

See Grading Rubrics for both Part A (Design) & Part B (Code)!