CS 135 - Project 1 -- Calculator Practice Program

INTRODUCTION

The purpose of Project 1 is to practice the skills that you are learning in Chapters 1-6 of the textbook.
In particular: C++ data types and operations (Chapter 2); Math expressions and Interactivity (Chapter 3); IF statements, menus, and switches (Chapter 4); WHILE and FOR loops (Chapter 5); and Functions (Chapter 6).

INSTRUCTIONS

Write a calculator practice program suitable for a seventh grade student.
You can use any of the code we developed in class, but you are instructed to write your own program.
When you are finished and want to test your program, document your tests by taking screenshots:
Use *Alt+PrtScn* to copy the active window, and paste it into an MS-Word document.
Your file shall be called ***screenshots.docx*** and include a sentence for each screenshot that tells what is going on in it: what the test was; what the result was; etc.
Your ***screenshots.docx*** should include at least 10 screenshots, along with associated description.
When you are ready to submit to Canvas, you need to zip up the ***Proj1*** folder and upload the resulting ***Proj1.zip*** file to the drop box.

Main Program Screen with Main Menu

```
WELCOME to the CALCULATOR PRACTICE PROGRAM.

This program allows you to practice your math skills.

Choose what you want to practice in the menu shown below.
-----------------------------------------------
                  MAIN MENU
-----------------------------------------------

a.   Addition                    (X+Y)
b.   Subtraction                 (X-Y)
c.   Multiplication              (X*Y)
d.   Division                    (X/Y)
e.   Powers & Roots
q.   Quit the program

-----------------------------------------------
Enter your choice [ a - e, q ]:
-----------------------------------------------
```

Powers & Roots Menu

```
Welcome to the Powers & Roots Menu.

This menu allows you to take powers and roots of a number.
-------------------------------------------------
                POWERS & ROOTS MENU
-------------------------------------------------

a.   Square a number             (X^2)
b.   Cube a number               (X^3)
c.   Raise to any power          (X^Y)
d.   Square root a number        (X^1/2)
e.   Cube root a number          (X^1/3)
f.   Take any root               (X^1/Y)
m.   Return to Main Menu
q.   Quit the program

-------------------------------------------------
Enter your choice [ a - f, m, q ]:
-------------------------------------------------
```

PROGRAM REQUIREMENTS

Implement the calculator as a menu program with prompts as necessary.
The menus shown above are specifications, that is, they should look exactly like those shown above.
Show the first Welcome menu on a cleared screen, and do **NOT** clear the screen for each subsequent menu.

For each computation in the Main Menu, before getting the numbers, ask the user if they want to perform:
- integer arithmetic (whole numbers input and whole number results); or
- floating point arithmetic (input numbers may contain a decimal, and output must contain a decimal).

However, since integer arithmetic doesn't make sense for roots and powers, that menu shall only do floating-point operations. Integer vs. floating-point will only be tested for: addition, subtraction, multiplication, and division.

Use stream manipulators to format your output. (Sec 3.7, p. 110)

An in-class demonstration is required for this project. Not giving a demo to the class will result in a grade of 0.

CODING CONVENTION

- Each file shall have a file header.
- Each function shall have a function header.
- The code shall be properly indented and commented.
- The code shall include the lines in the *pgm_template.cpp* at the end of the *main()* function that will output your name and the compilation date, and pause the program.
- The program shall demonstrate modular design, with a separate function for each math operation.
- Function recursion shall not be used.

PROJECT GRADING

A programming project is like a take-home test.

The student is given an extended amount of time, usually one or more weeks, in which to complete the program.

These items will be considered in grading the finished product:

1) Basics:
   (a) Is the project complete?
   (b) Is the code correctly compiled?
   (c) Does the program run by double-clicking?
   (d) Does the program run without crashing?
2) Did the student follow the written instructions?
   It is invalid to "reinterpret" or "change" the instructions in order to simplify the project.

SUBMISSION INSTRUCTIONS

Compile and test your code in the MinGW environment we have on campus. That is how it will be graded.

Create a folder named ***Proj1*** containing the following folders and contents:
Calculator, containing:
- *Calculator.exe*
- *Calculator.cpp*

The ***screenshots.docx*** file shall be in the main folder: ***Proj1***.

Zip up your folder and submit your ***Proj1.zip*** file to the Canvas Drop box.

Verification

To verify that your project is properly submitted, you can download your ZIP file from your Canvas.
Test your submission by performing the following procedure:

1. download your ZIP file from the Canvas drop box
2. unzip your ZIP file (extract its contents)
3. double-click on *Calculator.exe* to run it.
4. test the program to see if it behaves according to the specifications.

If the process (steps 1-4) doesn't work for you, it won't work for the grader. Fix it.