

## **Homework #7**

(C Programming for Beginners - OnLine)

**Note:** *Requires very basic understanding of pointers, lecture# 13*

Extend the simple calculator you wrote in Midterm, and make it more advanced by making it behave more like how the today's calculator user interface works. For example allow it to accept the input as an equation and print the results out to user.

**Details:** You need to write a program that when run, will clear the screen and asks the user to write the simple equation they want to evaluate. If the user enters a valid equation, it then displays the result. If user input is not valid, it displays an error message and asks for the correct equation. After the result is shown, it asks the user to enter another equation or press enter without any character to exit the program. The program continues infinitely until the user hits enter without entering anything else.

For simplicity, your program accepts only +, -, \* and / as operator. Only one operator is allowed per equation operating on two valid number operands.

### **Sample Run of the program:**

Welcome to <John Doe>'s Advanced Calculator, please enter a valid equation followed by enter :

12 + 12.00 ← (<user presses enter>)  
12.00 + 12.00 = 24.00

Please enter a valid equation and press enter or press enter to exit.  
\$5 \* 9.0 ← (<user presses enter>)

Not a valid equation.  
Please enter a valid equation and press enter or press enter to exit.  
2\*1 ← (<user presses enter>)  
2.00 + 1.00 = 2.00

Please enter a valid equation and press enter or press enter to exit.  
5 \* 9.0 + 3 ← (<user presses enter>)

Not a valid equation.  
Please enter a valid equation and press enter or press enter to exit.  
← (<user presses enter>)

Thank you for using <John Doe>'s Advance Calculator, good bye!  
Press enter to exit

**Note:**

- 1) Replace the <John Doe> with your name
- 2) You should allow only four operators (+, -, \*, /). And only one operator per equation.
- 3) Your program should allow input of integer or decimal numbers. The output should always be in decimals with two decimal digits as precision. A comma, or a \$ in a number is invalid
- 4) Give appropriate message when invalid menu choice is selected and re-display the prompt.
- 5) You should also appropriately catch the divide by zero issue and invalid inputs – e.g. if someone enters characters instead of numbers.
- 6) When user enters wrong value, give the appropriate message and ask for the correct values again. Repeat until valid values are entered.
- 7) Program only ends when user presses enter without writing any characters.