CETN 3010 PROGRAMMING IN C

ASSIGNMENT #1 – CONSOLE CALCULATOR

Due: Tuesday, Sept. 29, 2015 @ 10:30AM (ie. Before class begins)

General Requirements

REQ-001 Recreate the Included Calculator Program

Using the C programming language and Visual Studio, reverse engineer and re-create your own version of the accompanying program *Calculator.exe*. Your program should resemble and mimic exactly the functionality that is included in the sample Calculator application.

REQ-002 Naming your Project

You must name your Visual Studio project with the following naming convention.

*[Your Name]*_CETN3010_Assignment1

REQ-003 Code Requirements

Your recreation of this application must include the following techniques discussed in class:

1. Input and Output using printf() and scanf() functions.
2. Implementation of one of the looping techniques discussed in class. (There is one that is particularly appropriate for this application)
3. Appropriate use of arithmetic operators.
4. One example of a *switch* statement and at least one *if* statement.
5. Division by 0 is handled appropriately.

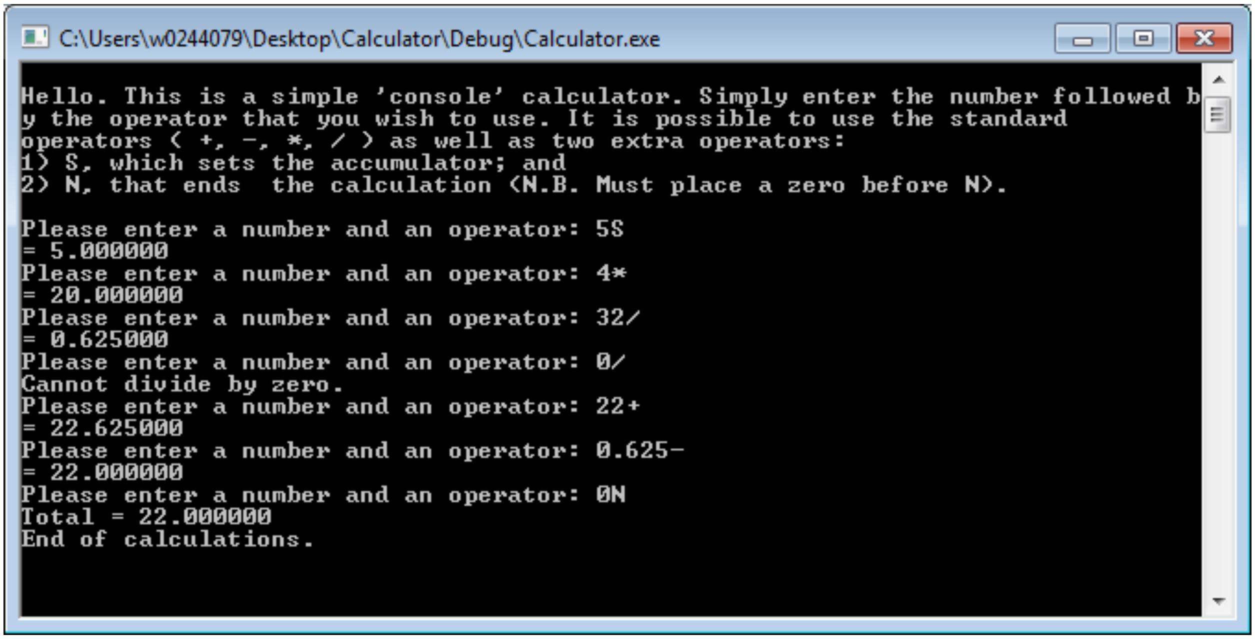
Submission via Source Control (Github)

You will submit this particular assignment via Github. Your submission will be time-stamped by Github as the latest commit you did to save your code. It is suggested that you commit often and confirm your final commit with your instructor.

Instructions

1. **Don't forget that a code review is part of this project. You will need to show your code to the instructor in class on the due date while going through an evaluation of the application's functionality. You will need to explain how the code works and complete the code review part of the rubric.**
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***

Sample Output



```
Hello. This is a simple 'console' calculator. Simply enter the number followed b
y the operator that you wish to use. It is possible to use the standard
operators ( +, -, *, / ) as well as two extra operators:
1) S, which sets the accumulator; and
2) N, that ends  the calculation (N.B. Must place a zero before N).

Please enter a number and an operator: 5S
= 5.000000
Please enter a number and an operator: 4*
= 20.000000
Please enter a number and an operator: 32/
= 0.625000
Please enter a number and an operator: 0/
Cannot divide by zero.
Please enter a number and an operator: 22+
= 22.625000
Please enter a number and an operator: 0.625-
= 22.000000
Please enter a number and an operator: 0N
Total = 22.000000
End of calculations.
```

	Developing	Competent	Professional	Mark Awarded
Console Input and Output	Developing (0-1 pts.) Little or no elements of this requirement are met.	Competent (2-3 pts.) Some elements of this requirement are met. Some errors exist.	Professional (4-5 pts.) All elements of this requirement are met.	
Arithmetic Operators	Developing (0-1 pts.) Little or no elements of this requirement are met.	Competent (2-3 pts.) Some elements of this requirement are met. Some errors exist. Not all operators have been implemented. Division by 0 check is not implemented.	Professional (4-5 pts.) All elements of this requirement are met. Division by 0 is handled appropriately by the application.	
Looping Structures	Developing (0-1 pts.) Little or no elements of this requirement are met.	Competent (2-3 pts.) An implementation of a looping structure exists but is not the best choice for the functioning of the application. Some error in the implementation exists.	Professional (4-5 pts.) All elements of this requirement are met. Appropriate looping structure is used. Users can perform as many calculations within the application as is needed.	
Conditional Statements	Developing (0-1 pts.) Little or no elements of this requirement are met.	Competent (2-3 pts.) Some elements of this requirement are met.	Professional (4-5 pts.) All elements of this requirement are met. An example of both a *switch* statement and an *if* statement are present in the application.	
Code Commenting	Developing (0 pts.) Little or no elements of this requirement are met.	Competent (1 pts.) Some elements of this requirement are met.	Professional (2 pts.) All elements of this requirement are met. Code is adequately commented with the programmer's intention.	
Formatting/ Readability	Developing (0 pts.) Little attention to correct spacing, indenting and adequate use of curly braces. Code very difficult to read	Competent (1-2 pts.) Some attention to correct spacing, indenting and adequate use of curly braces but could be improved. Code slightly difficult to read	Professional (3 pts.) Code is well spaced and attention is paid to the overall layout of the code. Spacing, indentation and correct use of curly braces are all implemented in a professional manner.	
Student Number:		**Student Name:**		Total: (/40)