Week 4 - Subroutines program

You will continue to develop the 2 programs started in the first assignment.

The goal for this week's program (which we will continue to build on in future assignments) is to create a modular version of the calculator program. The main routine should only consist of variable declarations and subroutine calls.

1. The calculator program will develop and use several functions and a subroutine:

Be sure to declare your 2 int variables in the main routine to hold the user's numbers and additional int and double variables to hold the respective answers

Call a single function (twice) that will ask for one of the int numbers to be used in the equations. Be creative how you ask the user for the first and second integer value!

Have separate functions to make each calculation (6 functions) and return the answer to main.

Now have a single output subroutine to produce output similar to the original program: operand1 operator operand2 = answer (italics indicate variables). You may have a second output subroutine to produce the floating point division equation.

Run your program twice with different integer values (be sure not to use zero as your second value) and supply both sets of Console output with your code.

Week 6 - Selection assignments/programs

For this programming assignment, you will use your previous code that implemented a tic-tac-toe game and a calculator. Correct any errors that were noted on the program or in the grade book.

For this and all future assignments, error-checking is a requirement.

1. The tic-tac-toe program will now number the 9 squares, so change your output routines accordingly.

1 | 2 | 3 etc. (4,5,6 is the second row and 7,8,9 the third)

Now ask the user, in a function, which square they want (1-9), and error check this value. Check to make sure that square is still available (otherwise output an error message) and place the user's mark 'X' in that square. This is easiest if you declare nine variables to hold the number or mark placed in a square. Now output the board with the user's mark in the correct square.

Repeat the process for a second player, who's mark is 'O'.

Run tic-tac-toe twice, once cleanly and once demonstrating your error checking and include both Console outputs.

2. For the calculator program the error checking will prevent either division from occurring if the second number is zero and will prevent the modulus operator from occurring if the second number is less than 1. All other operations should be performed as normal.

Again, run the calculator twice: once cleanly with a valid second number and once using zero as your second number.

Week 7 - Switch program

This program will modify your calculator program.

Instead of performing all 6 of the operations (+,-,\*,/,%, and float /) your program will now include a function that will ask the user which operation they wish to perform. Be sure to error check this to make sure the user specified a legitimate operator.

NOTE: Always error check any values a user inputs.

You will need a variable in main to hold the newly specified operator.

Now use a switch statement to determine which operation subroutine needs to be performed, execute that function and then output the result for that single operator.

Output 2 blank lines at the end of your program.