CSC 1710 LAB FOUR

VARIABLE TYPES & ARITHMETIC OPERATORS

DRS. TITUS AND WILLIAMS FALL 2020

THE ASSIGNMENT

In this lab you will investigate the output of arithmetic operations using `int` and `double` variable types. You will make predictions concerning the output from a number of expressions then will create a **single** program to test those expressions and make appropriate observations.

PREDICTING THE OUTPUT

Evaluate the expressions in the table below and fill in YOUR EXACT predictions of the values **stored in memory** in the second column. For the last two expressions, write C++ code for the stated expression in the first column, then put your prediction in the second column. You may use a hand-held calculator but remember to think like the C++ compiler! The following variables are declared:

```
int p=17, q=4, r=3;
double j=3.00, k=5.0;
```

Expression	Prediction	Program Output	Comment
p-q*r	5	5	Order of operations
p/r	5	5	divison
p%r	2	2	I found the remainder
p/q/r	1	1	read from left-right
q+r%p	7	7	Order of operations
q*j/p	0.70	0.705882	didnt include enough digits
p/q/j	1.33	1.33333	Same
p/j/q	1.41	1.41667	Same
k /=r/q	error	int	because you divide by zero
"j divided by the negative of q" Code: j/-q	-0.75	-0.75	division
"Area of a triangle w/base p, height r" Code: p·r/2	25	25	read left-Right

STOP HERE UNTIL YOU HAVE SHOWN YOUR PREDICTIONS TO THE INSTRUCTOR OR LAB ASSISTANT! DO NOT change your answers after this point!

WRITING CODE TO TEST OUTPUT

Do you find yourself wondering, “what output would I get if I did this . . . .?” For example, given variables `a`, `b`, `c`, `d`, and `x` declared below, we could use the following code to evaluate and output `c+a/b` and `a%d`:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   int a=10,b=2,c=4,d=7, x;
   x=c+a/b;
   cout << showpoint;
   cout << "c+a/b = " << x << endl;
   cout << "a%d = " << a%d << endl;
   return 0;
}
```

Notice that the expression can be output in the `cout` statement or can be assigned to a variable then the variable can be used in the output statement. Either approach is acceptable. NO other manipulators other than `showpoint` should be used.

Today you will write a similar program of your own to investigate the output of the different arithmetic expressions that you just evaluated by hand. **Once you finish your table of predictions**, create a `lab4` directory in your `csc1710` directory and create a new file for source code. Write a program that declares and initializes the variables `p`, `q`, `r`, `j`, and `k` as previously specified. Then create statements to compute the expressions and output their values. As in the example above, you may either assign the expression to a new variable of the same type then include that variable in your `cout` statement or you might include your calculations in your `cout` statement. Be sure to include a description of the calculation in the `cout` output string so that your output will be properly labeled. You might recall from class that an `int` divided by an `int` is an `int`, while, a `double` divided by an `int` is a `double` or a `double` plus an `int` is a `double`.

Caution: DO NOT assign expressions to the variables `p`, `q`, `r`, and `j` or you will change their values for future calculations! You may reassign a value to `k`.

Compile and execute your code. Make a note of the output for each in the 3rd column of your table and compare it with your predictions. If your prediction did not match the output EXACTLY, indicate in the Comments column what it was that you overlooked or at the time didn't understand that caused your prediction to be off. If your prediction does match the output EXACTLY, indicate the concept(s) that you applied correctly in making your prediction. Don't be afraid to admit that your prediction was incorrect--this is a learning experience and you will be graded on completeness, not on the correctness of your prediction!

Properly document your code. (COMMENTS!!) When you are finished with your program, turn in your chart and upload your source code. Your filename should include your **last name** and follow the guidelines for UNIX filenames (no spaces!).