Lab 3 - CMPS 1044, Computer Science I
Keyboard Arithmetic

Lesson objectives:

1. Practice of keyboard input and screen output
    a. User prompts for input
    b. Commented output for user
2. Various arithmetic problems
3. Transforming from `cout` to file output after debugging
4. Refresher on program documentation

Data can be provided to a program in 2 primary ways:

1. Interactively through the keyboard
2. Through a data file

In lab 2 you used a file. This week we will practice keyboard input. Type in the following program code in C++ then execute the code by pressing **CTRL+F5**. What happens? Why?

```
#include<iostream>
#include<fstream>

using namespace std;

int main()
{
      int num;
      cin >> num;
      cout << "The number entered is " << num << '\n';
      system("pause");
      return 0;
}
```

A blank screen appears, even though our program works. Without looking at the code, how is the user supposed to know to enter a number? We need a user prompt.

User Prompt: A user prompt is information displayed for the user giving instructions as to what they are to do. You need to add a user prompt telling the user to type in a number. Add the following line to the program above. Insert it immediately following the "`int num;`" statement. Execute again.

```
cout << "Please type in an integer, then hit the enter key.\n";
```

Arithmetic Operators: `+  -  *  /  %`

Let's practice with a variety of arithmetic operations using the value you type in for num.

Write a command to print out 5 sequential integers, including your entered number in the middle. That is, if you type in 9, your program should print out: 7 8 9 10 11

Add a second `cout` statement after the one already in your program. Type the following code and execute the program again.

```
cout << num - 2 << num - 1 << num << num + 1 << num + 2 << '\n';
```

What is wrong with this statement? Fix it by adding spaces between each number.

Integer division: When dividing 2 integers, the result is **ALWAYS** an integer. Add the following command to your program. Execute and type in an **ODD** number.

```
cout << num << " / 2 = " << num / 2 << '\n';
```

If you want the actual decimal value, you must ensure that one of your values is a real number. Change the 2 in the division to 2.0 then execute again. You can also **typecast** a variable so that it temporarily acts as if it is another variable type. In the above statement, typecast the variable in the division to a double by replacing it with either `(double)num` or `double(num)`.

Modulo (%) is the remainder function. It is used **only** for integer division. Add the following command to your program. Type in a large number (e.g. 4672) for input.

```
cout << "The remainder of " << num << " / 99 is " << num % 99 << '\n';
```

LAB 3 – Assignment

Modify the program you have been working on to include the following:

1. Add header of comments containing required information (as specified in the previous labs). Add at least 3 comments to the body of the program.
2. Add code to your program to accomplish the following tasks.
   a. Prompt the user to enter 2 different numbers. Call them `Number1` and `Number2`. (Don't forget to declare the 2 new integers at the beginning of the program.)
   b. Using output lines similar to those above, print a separate line to display an equation for each of the following operations:
      i. `Number1 + Number2`
      ii. `Number1 – Number2`
      iii. `Number1 * Number2`
      iv. `Number1 / Number2`
      v. `Number1 % Number2`

      Remember to typecast one of the variables for the division.
3. Execute your program entering 35 as `Number1` and 16 as `Number2`.
4. When the program is working correctly, change all `cout` statements to statements that will print to a file, as you learned in lab last week. To accomplish this, click the **Edit** menu, hover over **Find and Replace**, then click **Quick Replace**. You can also type **CTRL+H**. Once you have included the code for creating an output file, replace all instances of `cout` with your output file variable name.
5. Print out both the program code and the output file to turn in. SAVE your work before leaving the lab.