

CS 1428
Fall 2019
Gentry Atkinson
Lab 2

Introduction:

In last week's lab we learned how to define variables and how to use expressions to modify the values stored in those variables. Many times when we write code we need the execution of that code to depend on a defined condition. These techniques are called **Branching** because they produce code whose execution splits into different paths of execution based on conditions we define. C++ provides the **if**, **else if**, and **else** statements for branching.

The purpose of this lab is to familiarize you with Branching.

Directions:

1- Launch Code::Blocks and start a new file. Name it your_last_name_lab2.cpp.

2- Include the standard header for this lab:

```
//Your Name  
//CS1428 Fall 2019  
//Lab 2
```

3- Include the iostream standard library and start your main function:

```
#include <iostream>  
using namespace std;  
int main() {
```

4- Declare the following variables:

1. An integer called myCount.
2. A boolean called myCondition.
3. Assign appropriate values to myCount and myCondition.

5- Practice branching by copying the following code:

```
if(myCondition == true){                                //two equal signs!
    cout << "myCondition is true" << endl;
}
else {
    cout << "myCondition is false" << endl;
}

if(myCount < 10 || myCount > 99){
    cout << "myCount is not a double digit number" << endl;
}
else{
    cout << "myCount is a double digit number" << endl;
}
```

This block of code will execute the first action if the condition is true and it will execute the second block if that expression is false. Make sure you're using two equal signs rather than 1 (i.e. ==, not =).

6- Rewrite the following statement to accomplish the task described:

```
cout << "Please enter a number: ";
cin >> myCount;
cout<<endl;
if(/*enter a condition here using the % operator*/){
    cout << "This is an even number." <<endl;
}
```

7- Use your understanding of branching to implement the following pseudo-code. You can use a boolean 'or' (written ||, which is over the \ on your keyboard) to put several conditions into one **if** statement. The true branch will execute if any one condition is true

```
//DECLARE A CHAR TO HOLD SOME USER INPUT
//ASK THE USER TO INPUT A CHARACTER
//IF THE INPUT IS a, e, i, o, or u
    //OUTPUT "This character is a vowel."
//ELSE
    //OUTPUT "This character is not a vowel."
//PRINT: Thank you.
```

9- Do not copy the following code. Instead, predict the value of the int **outputValue** after the following code has executed. Print your prediction in the console with a **cout** statement.

```
bool a = true;
int b = 1, c = 2, outputValue = 0;

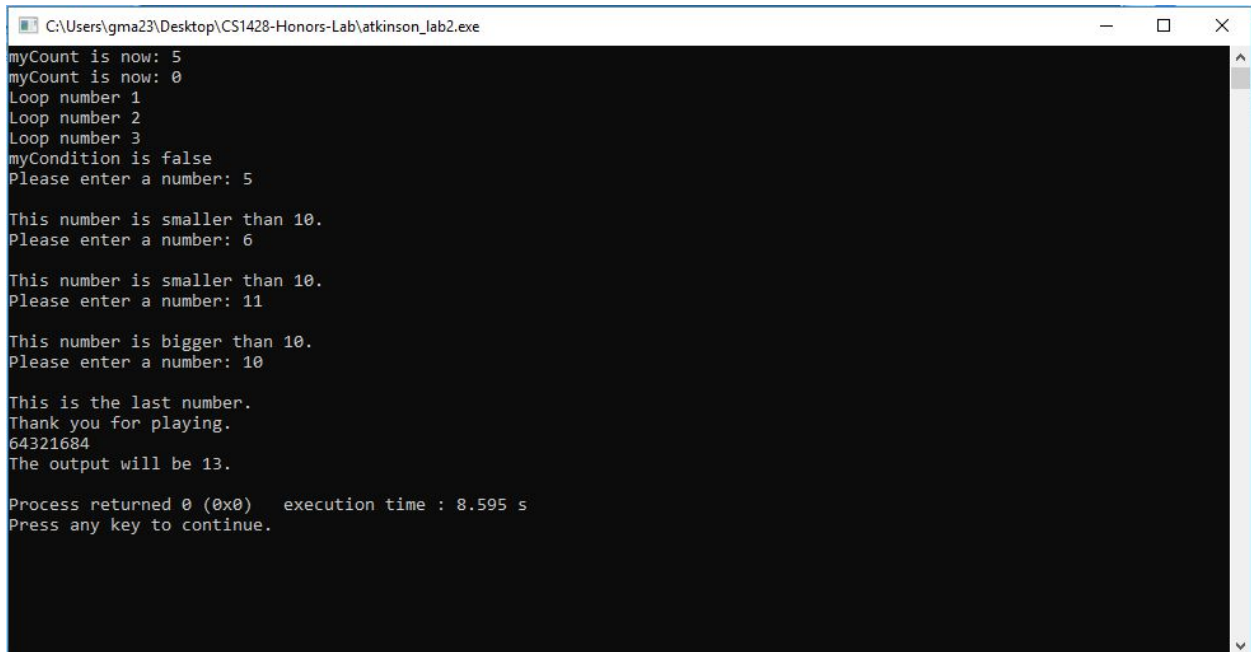
if(b <= c){
    a = false;
}
else{
    b = 3;
}

if(a){
    b = c;
}
else{
    c=b;
}

if(c==1){
    outputValue = 100 * c;
}
else {
    outputValue = 100 * b;
}
```

Notice that using one character variable names makes code hard to read and is generally a bad idea.

10- Save your work. Build and Run your code. Fix any errors. Your output should be something like this:



```
C:\Users\gma23\Desktop\CS1428-Honors-Lab\atkinson_lab2.exe
myCount is now: 5
myCount is now: 0
Loop number 1
Loop number 2
Loop number 3
myCondition is false
Please enter a number: 5

This number is smaller than 10.
Please enter a number: 6

This number is smaller than 10.
Please enter a number: 11

This number is bigger than 10.
Please enter a number: 10

This is the last number.
Thank you for playing.
64321684
The output will be 13.
Process returned 0 (0x0)   execution time : 8.595 s
Press any key to continue.
```

11- Submit your .cpp file through TRACS as an attachment. You can leave when you're done.