ES1036: Programming Fundamentals – Winter 2018

Lab02: Conditional statements in C++

Faculty of Engineering, Western University

Please remember your code files are due for submission on OWL at the end of your lab session; upload your files before leaving the lab.

A. Rationale and Background

In Lab 01, we used C++ expressions to perform simple calculations and input output statements. In this lab we will learn to write **conditional statements**. In real life we make decisions on conditions. For example **IF** you study hard, **THEN** you will be successful. We make similar decisions in C++ using the `if-else` structure. In this lab we will open the existing solution *ES1036Labs* and create the project *Lab02*.

B. How to Get Full Credit for this Lab

You will get credit for your lab when you demonstrate it to your TA, and get your TA's approval. During code-demonstration, your TA will ask you some questions on your code and request you to modify your code and/or write some other simple codes. When you finish working on Lab 02, you will be required to upload your .cpp files **(no other file types are accepted!)** to OWL after changing the file name by following the naming convention as suggested in Lab 00. A zero grade will be awarded for a **missing OWL submission** or **missing the lab demonstration session**.

- **Submission Instructions**

Number of files	:	**3**
Files to be submitted	:	cpp files in **E.1, E.2 and E.3**
Naming requirements	:	your_Western_user_name_lab02_q1.cpp, your_Western_user_name_lab02_q2.cpp, your_Western_user_name_lab02_q3.cpp

C. Pre-lab Questions

It is important that you read and understand the lab beforehand. Answering the following questions will help you get ready. You do not have to submit the pre-lab but be ready for a simple lab quiz.

1. Note that previously we have included the statement “using namespace std;” in our programs to use the cout, cin, and endl commands. Unfortunately, when using this statement, we also include a lot of overhead that is not being used. We have also seen that this statement can be omitted if we specify the scope of each command used using the scope resolution operator “::”, which is a time consuming solution. From now on, we will now emphasize which parts of the standard namespace we wish to include by using the following commands:

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;

int main() {
   cout << “We can eliminate the need for scope-declaration for cout, cin, or endl.\n”;
}
using std::cin;
using std::endl;
```

2. Explain how you would use the keywords `if` and `else`?
3. What is the purpose of the following program?

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;

int main()
{
        double marks;

        cout<<"Please enter your marks: ";
        cin>>marks;

        if(marks>50)
        {
                cout<<"Pass"<<endl;
        }
        else
        {
                cout<<"Fail"<<endl;
        }
        return 0;
}
```

4. Try to run the following code in Visual Studio or Xcode with a variety of inputs (listed below) and try to understand why some inputs produce “unreasonable” results. Run the code multiple times with the following inputs {2, 3, 49, 50, 65, 66, 97, 98, 127, 128, 255, 256, 9999999999, 97.9, and 98.2}.

```
#include <iostream>
#include <string>
#include <cmath>
using std::cout;
using std::cin;
using std::endl;

int main(){


      double doubleNumber = 0;
      cin >> doubleNumber;
      int intNumber = doubleNumber; //Squeeze a double into an int
      char character = intNumber;   //Squeeze an int into a char
      int intNumber2 = character;   //Get integer value of a char
      cout << "doubleNumber: " << doubleNumber << endl
                  << " intNumber: " << intNumber << endl
                  << " intNuber2: " << intNumber2 << endl
                  << " character( " << character << " )\n";

}
```

Note: C++ allows the programmer to cast a variable from one type to another, either explicitly or implicitly. However, some conversions are "unsafe", which means a value can be converted into a value of another type that does not equal to the original value. The above exercise showed some examples of unsafe type conversions. In most cases, you should avoid these kinds of conversions in your program.

D. Good Programming Practice

- Indent your code. You can let Visual Studio/Xcode indent the code for you (***Recall this from lab0 activities).*** You will be marked for commenting your code as well as appropriate indentation.

E. Lab Questions

E1. Number grade to letter grade conversion

Requirements:

Write a program that allows the user to input a number grade and convert it to a letter grade according to the specifications below.

Specifications:

1. The user will input a number between 50 and 100 inclusive; assume that the user will enter integer numbers only.

2. The program will output a letter grade determined by the numerical input as:
    - (95 <= A+ <= 100)
    - (90 <= A < 95)
    - (85 <= A- < 90)
    - (80 <= B+ < 85)
    - (75 <= B < 80)
    - (70 <= B- < 75)
    - (65 <= C+ < 70)
    - (60 <= C < 65)
    - (50 <= C- < 60)

3. **Hint: You should use an if-else statement to distinguish between the different letter categories.**

Sample Output

```
//First time running program
Input a numerical course grade (0.0 - 100.0):
75
The letter grade is B.

//Second time running program
Input a numerical course grade (0.0 - 100.0):
60
The letter grade is C.

//third time running program
Input a numerical course grade (0.0 - 100.0):
89.999
The letter grade is A-.
Press any key to continue . . .
```

E2. Quadratic equation

Requirements:

Write a C++ program to solve a quadratic with form $ax^2 + bx + c = 0$ following the specifications below. Recall that the quadratic equation is used to solve for *x*.

Specifications:

1. A header (example in sample output) will be printed to the console showing the student's name, lab number and question.
2. The program will ask the user to enter values for the variables a, b, and c. We can assume that the user will enter real numbers and no validation of these numbers is required.
3. If $a = 0$ the program must determine if b = 0 and c = 0.
    a. If b ≠ 0, it should print the single root.
    b. If b = 0 and c = 0, then it is a trivial solution and the program should print an appropriate message.
    c. If b = 0 and c ≠ 0, then there exists no solution and the program should print an appropriate message.

4. If $a \neq 0$, and if $b^2 - 4ac \geq 0$, the program should calculate the two real roots and display them using the formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

5. If $a \neq 0$, and if $b^2 - 4ac < 0$, it should calculate the two complex roots and display them using the formula:

$$x = -\frac{b}{2a} \pm j\,\frac{\sqrt{4ac - b^2}}{2a}$$

6. Roots should be displayed up to 4 decimal places.
    a. Use functions in `<iomanip>` to format your output to 4 decimal places.
    b. You must include the `<iomanip>` library file to use its functions.
    c. For example:

```
double x = 55/3.14;
std::cout << std::setprecision(4) << std::fixed << x << std::endl;
```

Sample Output:

```
//First run of the program
******************************************************
*               ES1036 Lab02    Q2                   *
*               Your Name                            *
******************************************************
This program solves a quadratic equation of the
form. ax^2 + bx + c = 0
Please input a: 1
Please input b: 1
Please input c: -2
Your equation has real roots: 1.0000 and -
2.0000.
Press any key to continue . . .

//Second run of the program
******************************************************
*               ES1036 Lab02    Q2                   *
*               Your Name                            *
******************************************************

This program solves a quadratic equation of the
form. ax^2 + bx + c = 0
Please input a: 0
Please input b: 0
Please input c: 1
A solution does not
exist!
Press any key to continue . . .

//Third run of the program
******************************************************
*              ES1036 Lab02    Q2                    *
*              Your Name                             *
******************************************************

This program solves a quadratic equation of the
form. ax^2 + bx + c = 0
Please input a: 1
Please input b: 2
Please input c: 2
Your equation has complex roots: -1.0000 + 1.0000i and -1.0000 -
1.0000i.
Press any key to continue . . .
```

E3. Encoding characters

Write a program that will encode a message using the ASCII character encoding scheme.

Character	A	B	C	D	E	F	G	H	I	J	K	L	M
ASCII	65	66	67	68	69	70	71	72	73	74	75	76	77

Character	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
ASCII	78	79	80	81	82	83	84	85	86	87	88	89	90

Character	a	b	c	d	e	f	g	h	i	j	k	l	m
ASCII	97	98	99	100	101	102	103	104	105	106	107	108	109

Character	n	o	p	q	r	s	t	u	v	w	x	y	z
ASCII	110	111	112	113	114	115	116	117	118	119	120	121	122

Specifications:

1. The program will encode the characters with a Caesar cipher; one of the earlier forms of encryption. Please see for details: https://en.wikipedia.org/wiki/Caesar_cipher
2. For this question, the user will enter a single character and the program will output an encoded character. The character must be encoded with a right shift of 7 (different shift value than the link posted).
    a. For example, if a user enters the character 'd', with an ASCII value of 100, the encoded character will be 'k', with an ASCII value of 107.
    b. If the user enters 'X', the output encoded character will be 'E'.
3. Your program must encode **any** upper or lowercase character specified in the chart above.
4. You must handle the cases where adding a shift to the character values produces a number outside of the range shown in the tables above. You may use an arithmetic expression using the modulo operator to "loop" around the table once exceeding an appropriate range, or you can handle the "edge cases" using if-else statements. Hint: IF adding an offset to the ASCII value moves the character outside of the range THEN perform a subtraction to the result to return to the start of the table. **Be prepared to discuss how the modulus operator can be used to accomplish this with your TA. Likewise, if you have questions using this operator, email your TA for extra clarification.**
5. If the user enters a character to encode that is outside of the ranges shown above, inform the user of the incorrect input and suggest they run the program again for another attempt.

Sample Output:

```
//First time running the program
Input a character to encode (a-z or A-Z): "
An incorrect character was input, please run the program again to input a new character.

//Second time running the program
Input a character to encode (a-z or A-Z): v
The encoded character is: c
```

F. Additional questions for practice

Level 1

1. Write a program that gets an integer number from the user and check if it is positive, negative or zero.
2. Write a program that prompts the user to enter an integer, and then show if it is an odd or even number. Assume that the user will enter integer numbers only.
3. Write a program that prompts the user to enter the age of a user and print " young" for less than 100 , "old" for 100 and "very old" for more than 100 years old.

Level 2

4. Write a program that prompts the user to enter number from the keyboard and display the square root of the number if it is between 16 and 38 inclusive otherwise it will display the half of the entered value.
5. A leap year is any year evenly divisible by 4 but not by 100, or if it is evenly divisible by 400. Write a program to tell whether a user input year is a leap year.
6. Write a program that prompts the user to enter three integers and finds the minimum of the three.

Level 3

7. Design a program that prompts the user to enter the year and month, and displays the number of days in the month. For example, if the user entered month 2 and year 2012, the program should display that February 2012 has 29 days. If the user entered month 3 and year 2005, the program should display that April 2005 has 30 days.
8. Write a program to compute the total cost to travel from Toronto to London, UK. The program should prompt the user for their preferred travel class (Flex, Latitude, and Business Class) and ask the user for the number of passengers. Per passenger fare is as follows
    1. Flex – 1000 CAD
    2. Latitude – 2000 CAD
    3. Business – 4000 CAD

    <u>Sample output should look like this.</u>

    ```
    Available travel classes
          1. Flex Class
          2.  Latitude class
          3. Business class

    Please enter your preferred class: 2
    Please enter the number of passengers: 3
    Total cost for your travel is 6000CAD.
    ```

9. Write a program that checks if the user-entered 3-digit number is divisible by 4, 5 and 9.

Sample output:

```
Please enter a 3 digit number: 475
The number is not divisible by 4.
The number is divisible by 5.
The number is not divisible by 9.
```

10. Write a program that asks the users about his/her height (m) and weight (kg) as an input and calculate the Body Mass Index and outputs the category of the person based on his/her BMI?

$$\mathrm{BMI} = \frac{\mathrm{mass(kg)}}{(\mathrm{height(m)})^2}$$

BMI Categories
Underweight = <18.5
Normal = 18.5-24.9
Overweight = 25.0-29.9
Obesity = 30.0->40