EE6411 Lab Sheet #1

Basic C++ Programs & Control Structures

Reiner Dojen[1]

[1]reiner.dojen@ul.ie

1 Objectives

- Learn how to write, compile and execute simple C++ programs.
- Use console input/output to print messages to the screen and to obtain values from a user.
- Learn how to use various C++ operators.
- Use various C++ control structures.

2 Preparation

Please make sure to install Microsoft Visual Studio Community (VS) on your system (you need to install the C/C++ package). If you haven't done so already, please work through E-Activity 1-1, which guides you through the process of setting up projects in VS and explains how to build and execute C++ programs. You should also be familiar on how to perform basic console input and output (if not, please work on E-Actiity 1-2).

3 Exercises

Ex1.1 Input/Output - Circle Calculations

Write a program that obtains the radius r of a circle (as a double) from the user. Subsequently, it should compute and print (using `cout`) the diameter ($2r$), the circumference ($2\pi r$) and the area (πr^2 - as the C++ language does not have a built-in exponentiation operator use $r * r$ instead of r^2). Use a constant variable to define $\pi = 3.14159$. Compile and run your program.

Ex1.2 Bitwise Operators

Write a program to read in two operands and display the bitwise AND (symbol '&'), OR (symbol '|'), XOR (symbol '^'), NAND (you need to combine other operators), NOR (you need to combine other operators) and the first operand shifted left (symbol '<<') and right (symbol '>>') by the number of bits expressed by the second of these operands. Use hexadecimal output for your results.

Example: Assuming the numbers 165 (0xa5) and 3 (0x3) have been entered, the following output is printed:

```
0xa5 AND 0x3 = 0x1
0xa5 OR 0x3 = 0xa7
0xa5 XOR 0x03 = 0xa6
0xa5 NAND 0x03 = 0xfffffffe
0xa5 NOR 0x03 = 0xffffff58
0xa5 Shift Right by 3 bits = 0x14
0xa5 Shift Left by 3 bits = 0x528
```

Ex1.3 Console Output - Size of Data Types

Write a program that prints out the size of all built-in types of the C++ language, i.e. for each type in C++ (`bool, char, wchar_t, char16_t, char32_t, short, int, long, long long, float, double, long double` (if you wish, you can also add the unsigned integer types - they should have the same sizes as their signed counterpart) and pointer type (use only a single pointer type, e.g. `int *`) print out their name and their size in bytes. Attention: for `char16_t` and `char32_t` you must use at least C++ 2011 standard (or newer) - on older compilers you may need to specify: `g++ -std=c++11`.

The size of a type can be obtained through the `sizeof` operator, e.g. `sizeof(int)` gives the size of the `int` type. Please note that the size is returned as type `size_t`, which is an unsigned integer type (usually `unsigned long`).

For each type print a message similar to "Size of char = 1 byte" (where the number is replaced with the obtained size). Use a separate line for each type. Compile your program and execute it.

Ex1.4 The `if-else` control structure

Use `if-else` construct(s) to determine the grade of a student: The user enters the obtained percentage grade and your program outputs the corresponding letter grade. Use the following percentage/grade mapping:

%	85	80	75	70	65	60	55	45	35	25	<25
Letter	A1	A2	B1	B2	B3	C1	C2	C3	D1	D2	F

Ex1.5 The `for` loop

Use a for loop to pring a table that converts °Fahrenheit to °Celsius. The relation between temperature in °C and °F is given by the formula:

$$^{\circ}C = \frac{5}{9} \cdot (^{\circ}F - 32)$$

Your table should be just two columns (no need for any any borders) with temperature in °F and °C for temperatures between 0 and 300°F in steps of 20°F. Compile and run your program. The following results should be obtained:

```
Fahrenheit  Celsius
0.00        -17.78
20.00       -6.67
40.00       4.44
60.00       15.56
80.00       26.67
100.00      37.78
120.00      48.89
140.00      60.00
160.00      71.11
180.00      82.22
200.00      93.33
220.00      104.44
```

```
240.00      115.56
260.00      126.67
280.00      137.78
300.00      148.89
```

Ex1.6 A variable `foor` loop

Modify the previous exercise: Rather than using fixed limits (0-300°F) and step size (20°F), prompt the user to enter the range of temperature in °F and also for the step size (in °F)

Ex1.7 The `while` loop

Write a program that asks the user to enter a positive integer number. Using a `while` loop, count how often the entered number can be divided by 2 before the limit of 100 is reached. For example, if the user enters the number 750 then:

```
750 ÷ 2 = 375 (> 100)
375 ÷ 2 = 187 (> 100)
187 ÷ 2 = 93 (< 100)
```

Thus, the number 750 can be divided twice by 2 before the limit 100 is reached.

Ex1.8 The `do-while` loop

Use a `do-while` loop to create a program that:

1) Asks the user to enter a postive integer number.
2) If the numbers is > 0, tell the user whether the entered number was even or odd.
3) Repeat steps 1-2 unless the user enters a value ≤ 0 - in this case exit the loop and terminate the program.

Ex1.9 The `switch` statement

Print the following menu to the screen:

```
1) Addition
2) Subtraction
3) Multiplication
4) Division
5) Exit
```

Ask the user to enter a number between 1 and 5. Use a `switch` statement to react to the input. For each input 1-4 read in two numbers, perform the corresponding calculation and print the result. Repeat the program until the user chooses 5.

Ex1.10 Another `switch` statement

Redo the previous exercise - however, this time use a `char` variable for user menu choice. Use the following menu:

```
+) Addition
-) Subtraction
*) Multiplication
/) Division
x) Exit
```

As `char` variables are compatible with integers, they can also be used in a `switch` statement. Also, make sure to use character constants (e.g. `'+'`) in the `case` clauses of your `switch` statement.