# Lab manual 1

Programming language concepts

Lecturer: sayed najmuddin”sadaat”

1. Introduction

In this practical you learn basics of C ++, you will also create and run your first C ++ app, hello world,

Variables, data types and other basics activity.

A simple c ++ program structure to display “hello world”

// header file for input output functions

#include <iostream>

Using namespace std;

// main function where the execution of program begins

Int main (){

Cout<< “hello world “; // prints hello world

Return

}

1. // simple C ++ program to display “hello world”: this line is a comment line. A comment is used to display additional information about the program. A comment does not contain any programming logic. When a comment is encountered by a compiler, the compiler simply skips that line of code. Any line beginning with ‘//’ without quotes or in between /\*…\*/ in c ++ is comment.
2. # include: in c ++, all lines that start with pound (#) sign are called directives and are processed by preprocessor which is a program invoked by the compiler. The # include directive tells the compiler to include a file and #include <iostream>. It tells the compiler to include the standard iostream file which contains declarations of all the standard input/output library functions.
3. Using namespace std: This is used to import the entirely of the std namespace into the current namespace of the program. The statement using namespace std is generally considered a bad practice. When we import a namespace we are essentially pulling all type definitions into the current scope. The std namespace is huge. The alternative to this statement is to specify the namespace to which the identifier belongs using the scope operator (::) each time we declare a type.
4. Int main (): this line is used to declare a function named “main” which returns data of integer type. A function is a group of statement that are designed to perform a specific task. Execution of every c++ begins with the main () function, no matter where the function is located in the program. So, every c++ program must have a main () function.
5. {and}: the opening braces‘{‘indicates the beginning of the main function and the closing braces ‘}’ indicates the ending of the main function. Everything between these two comprises the body of the main function.
6. Std::cout <<”hello world”;: this line tells the compiler to display the message “hello world” on the screen.

This line is called a statement in c++. Every statement is meant to perform some task. A semi colon ‘;’ is used to end the statement. Semi colon character at the end of the statement is used to indicate that the statement is ending there. The std:: cout is used to identify the standard character output device which is usually the desktop screen. Everything followed by the character “<<” is displayed to the output device.

1. Return 0; this is also a statement. This statement is used to return a value from a function and indicates the finishing of a function. This statement is basically used in function to return the return the result of the operations performed by the function.
2. Indentation: as you can see the cout md the return statement have been indented or moved to the right side. This is done to make the code more readable. In a program as hello world, it does not hold much relevance, but as the programs become more complex, it makes the code more readable, less error- prone. Therefore, you must always use indentations and comments to make the code more readable.

C++ identifiers

A C++ identifiers is a name used to identify a variable, function, class, module, or any other user-defined item. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by the zero or more letters, underscores, and digits (0 to 9).

C++ does not allow punctuation characters such as @, $, and % within identifiers. C++ is a case sensitive programming language. Thus, manpower and manpower are two different identifiers in c++. Here are some examples of acceptable identifiers-

Mohd zara abc move\_ name a\_123 myname50 \_temp j a23b9 retval

C++ keywords

The following list shows the reserved words in c++. These reserved words may not be used as constant or variable or any other identifiers names.

Asm else new this

Auto enum operator throw

Bool explicit private true

Break export protected try

Case extern public typedef

Catch false register typeid

Char float reinterpret\_ cast typename

Class for return union

Const friend short unsigned

Cons\_cast goto signed using

Continue if sizeof virtual

Default inline static void

Delete int static\_cast volatile

Do long struct wchar\_t

Double mutable switch while

Dynamic\_cast namespace template

Basic built in data types in C ++

Char for character storage (1 byte)

Char a = ‘A’; // character type

Int for integral number (2 bytes)

Int a = 1; // integer type

Float single precision floating point (4 bytes)

Float a = 3.14159 //floating point type

Double double precision floating point numbers (8 bytes)

Double a = 6e-4; // double type (e is for exponential)

Bool Boolean (true or false) bool a = true;

Void without any value

Wchar\_ t wide character

Modifiers in c ++

In c ++ special words (called modifiers) can be used to modify the meaning of the predefined built-in data types and expand them to a much larger set. There are four datatype modifiers in c++, they are: long, short, signed, unsigned

The above mentioned modifiers can be used along with built in datatypes to make them more precise and even expand their range.

Below mentioned are some important points you must know about the modifiers, long and short modify the maximum and minimum values that a data type will hold. A plain int must have a minimum size of short.

Size hierarchy: short int <int<long int

Size hierarchy for floating point numbers: float< double<long double

Long float is not a legal type and there are no short floating point numbers.

Signed types includes both positive and negative numbers and is the default type.

Unsigned, numbers are always without any sign that is always positive.

Exercise-1 with solution: write a c++ program to print ‘hello’ on screen and then print your name on a separate line.

#include<iostream>

Using namespace std;

Int main () {

Cout <<”hello dear “<<end 1 <<“Aziz Ahmad Afzali “;

Return 0;

}

Sample solution using input from the user:

#include<iostream>

Using namespace std;

Int main () {

String fname [10], 1 name [10];

Cout<<”Aziz Ahmad: “;

Cin >> fname;

Cout<<”Aziz Malik: “;

Cin >> 1 name;

Cout<<”hello dear “<< end1 << fname << sname;

Return 0;

}

# INCLUDE <IOSTREAM> DISPLAY CORRESPONDING ASCII value FOR THE GIVEN CHARACTER

In this program, we will read a single character from the keyboard and then display its equivalent ASCII value.

ASCII means American standard code for information interchange. For every character, there is an equivalent ASCII value like ASCII value for ‘A’ is 65 and ‘a’ is 97. Logic of this program is so simple. Here, we are converting character value to the integer value. So, it will automatically convert character to ASCII value. See the following program.

Using namespace std;

Int main ()

{

Char ch;

Int a;

Cout<< “string f name\n”;

Cin.get (ch);

a=ch; // character to integer

Cout <<”ASCII value for “<<ch<<” is “<<a;

Return 0;

}

Output of this program is

Enter any character a

ASCII value for a is 97