



By Dr. Samaher Hussein Ali

College of Information Technology, University of Babylon, Iraq

Samaher_hussein@yahoo.com

27 November 2012

Introduction

This course will cover an introduction to programming in C++. The programming environment used is C++.

The programming language C++ was originally developed in the 1980s to support the development of large software projects, including those using object oriented methods. The name signifies the evolution from C; the increment operator "++" signifies that the new language is the old one with additions; indeed C++ is a superset of C (there are a few exceptions). There is now an ANSI standard for C++.

There are many compilers and environments available for turning C and C++ programs into executable programs. These range from the simplistic to the supportive. The most simple systems require the programmer to use a text editor to enter all the code, the code is written to an appropriately name file, compiled, then linked to necessary libraries and execution of the code can then follow. The user corrects mistakes returning to the editor. More supportive environments tend to be integrated, with some automatic generation of code; the available libraries may be very extensive. Therefore, The following topics will be covered during the course of lectures

Topics

- □ Introduction of computer, software, software application
- □ Programming languages
- □ Compiler related with c++
- Overview of a program and determine
- □ The features of good program from the user' view
- □ The features of good program from the programmer' view
- □ The levels of programming languages
- Discuss the Algorithms and Flowchart
- Discuss Input and output(i.e., commands, Variables, Constants)
- □ Types and operators (Mathematical and Logical Operation)
- □ Iteration Statement (For, While, Do..While), Condition Statement(If, Switch)
- Functions
- 🛛 Arrays (ID, 2D)
- Strings
- Pointers
- □ Files and type of Files

❑ Structures

Computer

an electronic device which is capable of receiving information (data) in a particular form and of performing a sequence of operations in accordance with a predetermined but variable set of procedural instructions (program) to produce a result in the form of information or signals.

Computer is derived from the word compute which means compute or calculating in right manner. it can acess, store, and retrived the large amount of data without intervention of human.

Main Parts Of a Computer

- Processor (CPU) The part of the computer that handles all processes and instructions supplied by memory unit (RAM)
- Memory (RAM) Random Accessed Memory. The area of the computer that holds the instructions (processes) and information system gives it. When you turn the computer off, everything in RAM disappears.
- Read Only Memory (ROM) A chip or disk that holds information that cannot be changed. Ex CD-ROM, DVD-ROM
- Disk drive A mechanical device that you use to transfer information back and forth between the computer's memory and a disk.
- Floppy disk (3.5") A magnetically coated disk on which information can be stored and retrieved. Capacity is
 1.44MB requires a Floppy-Drive
- Zip disk A magnetically removable coated disk on which information can be stored and retrieved. Capacity is
 100 or 250MB or more requires a zip-drive

Computer

Hard disk - A large capacity storage area that offers access to store and retrieve information, very slow compare to RAM.

□ Monitor - A screen that displays the information in the computer.

□ Keyboard - A device used to enter data and issue commands to the computer.

Printer or Scanner - A devices that help to make hard copies o scan documents into the computer.

□ Mouse - A small, hand-held device used to control the pointer on the screen.

□ Software - Instructions that tell your computer how to perform a task.

Software is stored on the disks in program files. Software cannot be seen or touched. There are two main kinds of software:

Application software (Program): Software that does specific task, such as word processing. (Word, Power Point)

System Software: Software that the computer system or OS operating system.

Operating System (OS) - software that acts as a link between you, application software (programs), and hardware.

As a result, any computer consist of two main parts are Hardware and Software

Data, Information and Algorithm

What is Data?

- □ Collection of data objects and their attributes
- □ An attribute is a property or characteristic of an object

Examples: eye color of a person, temperature, etc.

□ A collection of attributes describe an object.

Object is also known as record, point, case, sample, entity, or instance

What is Information?

□ Collection of organization data or preparing data to take the decision , where

Data + Processing = Information

Example: book is data while my book is information.

What is Algorithm?

□ Collection of steps require to solve problem and it must have the following conditions:

The Conditions of Algorithm

- 1. "Keep it simple, Stupid"- Don't use fancy features when simple ones suffice.
- 2. "Read the fascinating manual"- Most complaints from the compiler can be solved by reading the book. Logical error something else.
- 3. "Make your documentation short but sweet"- Always document your variable declarations, and tell what each subprogram does.
- 4. "Every subprogram should do something and hide something"- If you cannot concisely explain what your subprogram does, it shouldn't exist. This is why I write the header comments before I write the subroutine.
- 5. "Program defensively"- Add the debugging statements and routines at the **begging**, because you know you are going to need then later.
- 6. "Good program is a pretty program"- Remember that you will spend more time reading your programs than we will perfect shuffles.

Programming Languages

What is programming language ?

□ A programming language is an artificial <u>language</u> designed to communicate <u>instructions</u> to a <u>machine</u>, particularly a <u>computer</u>. Programming languages can be used to create <u>programs</u> that control the behavior of a machine and/or to express <u>algorithms</u> precisely. A programming language is usually split into the two components of <u>syntax</u> (form) and <u>semantics</u> (meaning).

□ The term *programming language* usually refers to <u>high-level languages</u>, such as <u>BASIC</u>, <u>C</u>, <u>C++</u>, <u>COBOL</u>, <u>FORTRAN</u>, <u>Ada</u>, and <u>Pascal</u>. Each <u>language</u> has a unique set of <u>keywords</u> (words that it understands) and a special <u>syntax</u> for organizing <u>program</u>



27 November 2012

instructions.

Dr. Samaher Hussein Ali

Notes of Lecture 1

Programming Languages

□ Machine Language : The lowest-level programming language (except for computers that utilize programmable microcode) Machine languages are the only languages understood by computers. While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers. Programmers, therefore, use either a high-level programming language or an assembly language. An assembly language contains the same instructions as a machine language, but the instructions and variables have names instead of being just numbers. □ Assembly Language: A programming language that is once removed from a computer's machine language. Machine languages consist entirely of numbers and are almost impossible for humans to read and write. Assembly languages have the same structure and set of commands as machine languages, but they enable a programmer to use names instead of numbers.

□ High-Level Language: A programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of <u>computer</u>. Such <u>languages</u> are considered high-level because they are closer to human languages and further from <u>machine languages</u>. In contrast, <u>assembly languages</u> are considered low-level because they are very close to machine languages.

The main advantage of high-level languages over <u>low-level languages</u> is that they are easier to read, write, and maintain. Ultimately, programs written in a high-level language must be translated into machine language by a <u>compiler</u> or <u>interpreter</u>.

Compiler

□ The main advantage of high-level languages over <u>low-level languages</u> is that they are easier to read, write, and maintain. Ultimately, programs written in a high-level language must be translated into machine language by a <u>compiler</u> or <u>interpreter</u>.

What is Compiler?

□ A program that translates <u>source code</u> into <u>object code</u>. The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the <u>instructions</u>. Thus, a compiler differs from an <u>interpreter</u>, which analyzes and <u>executes</u> each line of source code in succession, without looking at the entire program. The advantage of interpreters is that they can execute a program immediately. Compilers require some time before an executable program emerges. However, programs produced by compilers <u>run</u> much faster than the same programs executed by an interpreter.

Every <u>high-level programming language</u> (except strictly interpretive languages) comes with a compiler. In effect, the compiler is the <u>language</u>, because it defines which instructions are acceptable.

Because compilers translate source code into object code, which is unique for each type of <u>computer</u>, many compilers are available for the same language.

Compiler



Dr. Samaher Hussein Ali

Notes of Lecture 1