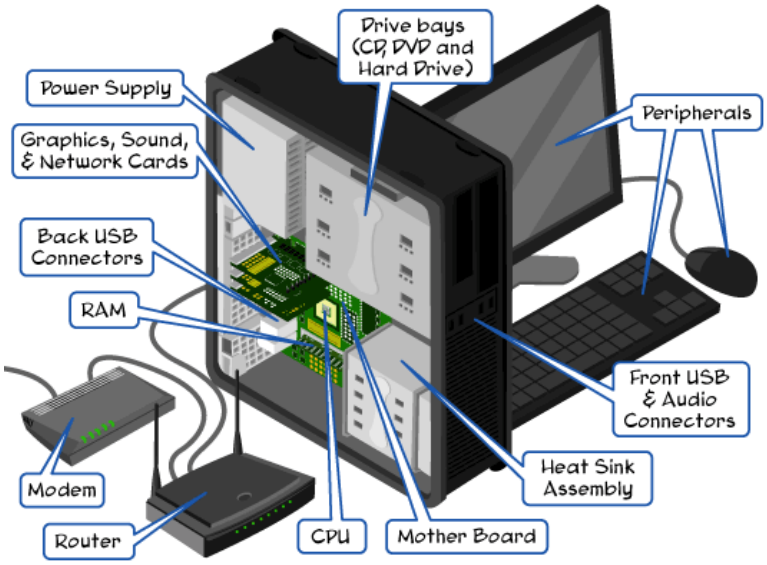
Lecture 1

computer:

is a device that accepts information (in the form of digitalized data) and manipulates it for some result based on a program or sequence of instructions on how the data is to be processed. Complex computers also include the means for storing data (including the program, which is also a form of data) for some necessary duration. A program may be invariable and built into the computer (and called logic circuitry as it is on microprocessors) or different programs may be provided to the computer (loaded into its storage and then started by an administrator or user). Today's computers have both kinds of programming.

The process of developing and implementing arious sets of instructions to enable a computer to do a certain task. These instructions are considered computer programs and help the computer to operate smoothly.



What is software and hardware?

Computer software, or simply software, is that part of a computer system that consists of encoded information or computer instructions, in contrast to the physical hardware from which the system is built.

Computer software includes computer programs, libraries and related non-executable data, such as online documentation or digital media. Computer hardware and software require each other and neither can be realistically used on its own.

software, is a collection of computer programs and related data that provide the instructions telling a computer what to do and how to do it.

Computer hardware (or simply hardware in computing contexts) is the collection of physical elements that constitutes a computer system. Computer hardware is the physical parts or components of a computer, such as the monitor, keyboard, computer data storage, hard disk drive(HDD), graphic cards, sound cards, memory (RAM), motherboard, and so on, all of which are tangible physical objects.[1] By contrast, software is instructions that can be stored and run by hardware..

Procedural programming: is derived from structured programming, based upon the concept of the procedure call. Procedures, also known as routines, subroutines, or functions (not to be confused with mathematical functions, but similar to those used in functional programming), simply contain a series of computational steps to be carried out. Any given procedure might be called at any point during a program's execution, including by other procedures or itself.

The history of C++:

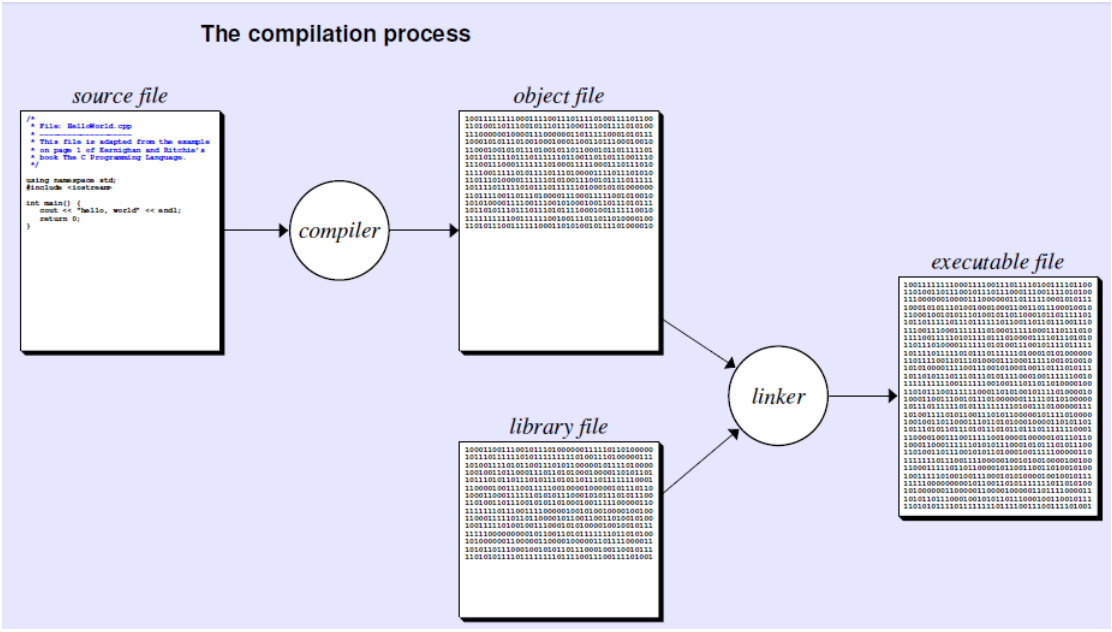
In the early days of computing, programs were written in machine language, which consists of the primitive instructions that can be executed directly by the machine. Programs written in machine language are difficult to understand, mostly because the structure of machine language reflects the design of the hardware rather than the needs of programmers. Worse still, each type of computing hardware has its own machine language, which means that a program written for one machine will not run on other types of hardware. In the mid-1950s, a group of programmers under the direction of John Backus at IBM had an idea that profoundly changed the nature of computing. Would it be possible, Backus and his colleagues wondered, to write programs that resembled

the mathematical formulas they were trying to compute and have the computer itself translate those formulas into machine language? In 1955, this team produced the initial version of FORTRAN (whose name is a contraction of formula translation),
which was the first example of a higher-level programming language. Since that time, many new programming languages have been invented, most of which build on previous languages in an evolutionary way. C++ represents the joining of two branches in that evolution. One of its ancestors is a language called C, which was designed at Bell Laboratories by Dennis Ritchie in 1972 and then later revised and standardized by the American National Standards Institute (ANSI) in 1989. But C++ also descends from a family of languages designed to support a different style of programming that has dramatically changed the nature of software development in recent years.

The most important thing while learning C++ is to focus on concepts. The purpose of learning a programming language is to become a better programmer; that is, to become more effective at designing and implementing new systems and at maintaining old ones.

The compilation process:

When you write a program in C++, your first step is to create a file that contains the text of the program, which is called a source file. Before you can run your program, you need to translate the source file into an executable form. The first step in that process is to invoke a program called a compiler, which translates the source file into an object file containing the corresponding machine-language instructions. This object file is then combined with other object files to produce an executable file that can be run on the system. The other object files typically include predefined object files called libraries that contain the machine-language instructions for various operations commonly required by programs. The process of 6 Overview of C++ combining all the individual object files into an executable file is called linking. The steps in the compilation process are illustrated in the Figure bellow.


The compilation process
source file
object file
compiler
linker
library file
executable file