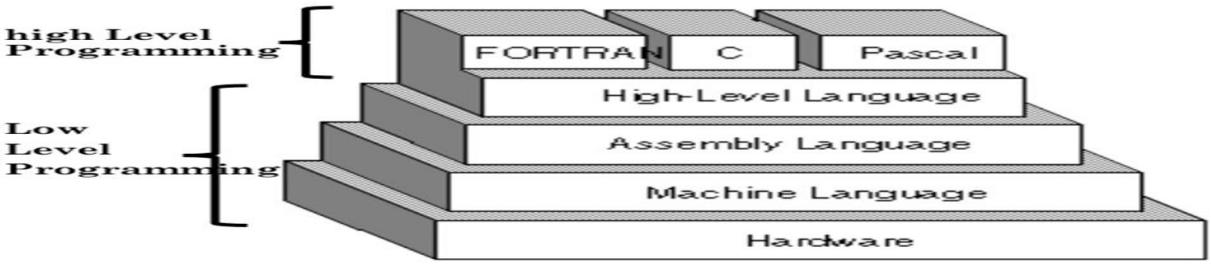
CHAPTER SEVEN

Programming Language

Programming languages:

Programming languages provide the basic building blocks for all systems and application software.

Programming languages allow people to tell computers what to do and are the means by which software systems are developed, we will describe the two-levels of programming languages:



Low level languages:

1 – Machine language:

Is the lowest-level computer language, consisting of the internal representation of the instructions and data. This machine code-the actual instructions understood and directly executable by the CPU is composed of binary digits. Machine language is the only

programming language that the machine actually understands, therefore, machine language

is considered the first-generation language. All other languages must be translated into machine language before the computer can run the instructions because computer's CPU is capable of executing only machine language programs.

Machine language is extremely difficult to understand and use by programmers. As a result, increasingly more user-friendly languages have been developed.

These user oriented languages make it much easier for people to program. But they are impossible for the computer to execute without first translating the program into machine language.

The set of instructions written in a user-oriented language is called a source program.

The set of instructions produced after translation into machine language is called the object program.

Programming in a higher-level language (i.e., a user oriented language) is easier and less time consuming but additional processor time is required to translate the program before it can be executed.

2 – Assembly language:

Assembly languages are considered second-generation languages it is more user-friendly because it represents machine language instructions and data locations in primary storage by using mnemonics, which people can more easily use.
Compared to machine language, assembly language eases the job of the programmers.

Translating an assembly language program into machine language is accomplished by system software program called an assembler.

High level languages:

1 – Procedural languages:

Called third-generation language

• Procedural language are much closer to natural language (the way we talk) and therefore, are easier to write, read.

• Procedural language use common words rather than abbreviated mnemonics.

• There are three examples of procedural languages FORTRAN, COBOL, and C.

2 – Nonprocedural languages:

- Called fourth-generation language.
- They can be used by non-technical users to carry out specific functional tasks.
- These languages simplify the programming process as well as reduce the number of coding errors.
- They are common in database applications as query languages, report generators.

3-Natural languages:

- Are called fifth –generation languages or" intelligent language".
- They are use mnemonics and tables.
- Most of these languages are still experimental because the programs that are translate natural language into machine–readable form are extremely complex and require a large amount of computer resources.

Newer programming languages:

1 – Visual programming languages:

- Are used within graphical environment.
- Are using a mouse, icons, and symbols on screen.
- Visual basic and visual C++ are examples of visual programming languages.

2 – Hypertext markup language (HTML):

- Is an approach to data management in which data are stored in a network of nodes connected by links(called hyperlinks).
- Users can access data through an interactive browsing system.
- The combination of nodes, links, and supporting indexes for any particular topic is a hypertext document.
- A hypertext document may contain text, images, and other types of information such as data files, audio, and video.
- World Wide Web (www) uses HTML for creating and recognizing hypertext document.

3 – Object-Oriented programming languages:

- Object-Oriented Programming (OOP) languages are based on the idea of taking a small amount of data and instructions about what to do with that data and putting both of them together into what is called an object.

- C++ and JAVA are examples of OOP languages.