Name: محمد سعد أحمد حسين علي

B.N: 15

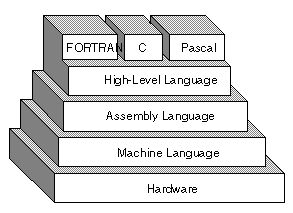
TOPIC: Programming languages

GitHub Link:

* Programming Languages
* Introduction

A programming language is a vocabulary and set of grammatical rules for instructing a [computer](https://www.webopedia.com/TERM/C/computer.html) or computing device to perform specific tasks. The term programming language usually refers to [high-level languages](https://www.webopedia.com/TERM/H/high_level_language.html), such as [BASIC](https://www.webopedia.com/TERM/B/BASIC.html), [C](https://www.webopedia.com/TERM/C/C.html), [C++](https://www.webopedia.com/TERM/C/C_plus_plus.html), [COBOL](https://www.webopedia.com/TERM/C/COBOL.html), [Java](https://www.webopedia.com/TERM/J/Java.html), [FORTRAN](https://www.webopedia.com/TERM/F/FORTRAN.html), [Ada](https://www.webopedia.com/TERM/A/Ada.html), and [Pascal](https://www.webopedia.com/TERM/P/Pascal.html).

Each programming language has a unique set of keywords (words that it understands) and a special [syntax](https://www.webopedia.com/TERM/S/syntax.html) for organizing program



* ***High-Level Programming Languages***

High-level programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands, called [*machine languages*](https://www.webopedia.com/TERM/M/machine_language.html). Each different type of [CPU](https://www.webopedia.com/TERM/C/CPU.html) has its own unique machine language.

Lying between machine languages and high-level languages are languages called [assembly languages](https://www.webopedia.com/TERM/A/assembly_language.html). Assembly languages are similar to machine languages, but they are much easier to program in because they allow a [programmer](https://www.webopedia.com/TERM/P/programmer.html) to substitute [names](https://www.webopedia.com/TERM/N/name.html) for numbers. Machine languages consist of numbers only.

Lying above high-level languages are languages called [*fourth-generation languages*](https://www.webopedia.com/TERM/F/fourth_generation_language.html) (usually abbreviated *4GL*). 4GLs are far removed from machine languages and represent the class of computer languages closest to human languages.

* ***Converting to Machine Language***

Regardless of what language you use, you eventually need to [convert](https://www.webopedia.com/TERM/C/convert.html) your program into machine language so that the computer can understand it. There are two ways to do this:

1) [Compile](https://www.webopedia.com/TERM/C/compile.html) the program.  
2) [*Interpret*](https://www.webopedia.com/TERM/I/interpreter.html) the program.

***Recommended Reading:****See*[*compile*](https://www.webopedia.com/TERM/C/compile.html)*and*[*interpreter*](https://www.webopedia.com/TERM/I/interpreter.html)*for more information about these two methods.*

The question of which language is best is one that consumes a lot of time and energy among computer professionals. Every language has its strengths and weaknesses. For example, FORTRAN is a particularly good language for processing numerical [data](https://www.webopedia.com/TERM/D/data.html), but it does not lend itself very well to organizing large programs. Pascal is very good for writing well-structured and readable programs, but it is not as flexible as the C programming language. C++ embodies powerful [object-oriented](https://www.webopedia.com/TERM/O/object_oriented.html) [features](https://www.webopedia.com/TERM/F/feature.html), but it is complex and difficult to learn.

## *The Top Programming Languages?*

According to IEEE Spectrum's interactive ranking, Python is the top programming language of 2017, followed by C, Java and C++. Of course, the choice of which language to use depends on the type of computer the program is to [run](https://www.webopedia.com/TERM/R/run.html) on, what sort of program it is, and the expertise of the programmer.

* The four different programming paradigms

According to the Merriam-Webster’s Collegiate dictionary, a paradigm is “A philosophical and theoretical framework of a scientific school or discipline within which theories, laws, and generalizations and the experiments performed in support of them are formulated”. In essence, we are in reference to patterns and models. With that said, there are four major types of programming paradigms; namely: **Imperative**, **Functional**, **Logical** and **Object-Oriented**.

The imperative programming paradigm was one of the earliest and was developed using machine-language. It is also the basis on which all hardware is implemented. This was further confirmed by a published journal by Laird (2009) in which he stated that “In **imperative programming**, statements are instructions at the native machine-level, and they contain states and variables that point right to the memory.”  Some of its merits include but not limited to: close to the machine – fast execution time, more efficient. Whilst, on the other hand, its weaknesses are order sensitive and the limitation of abstraction. **Functional programming** is essentially less complex and offers better readability than imperative.

This paradigm is thought of to have been originated from a mathematical discipline. Merits include: compare to imperative it has a higher level of abstraction, is not tied to dependency. Some of the weakness include: less efficiency, troubleshooting variables or it’s sequential activities are better handled in both Object-Oriented or imperatively. Unlike the other major paradigms, the **Logical paradigm** is vastly different in that it focusses primarily on predicate logic — relation. This is also a vital part of the logic circuit of a computer. Merits of this paradigm include: “problems are solved by the system and Proving the validity of a given program is simple” (the University of Central Florida. Finally, the **Object-Oriented paradigm** also known as O-O focusses on objects, it’s representation and the behavior they exhibit; rather than variables of the imperative paradigm. O-O borrows from all the major paradigms.

* Screenshots of code:

