

**essay**

# The Five Generations of Programming

## First Gen

The first generation of programming languages is what is called 'machine code'. This in essence is a series of 1's and 0's, and is not human-readable. It is however readable by computers, comprising of basic sequences understandable by other components of the computer to perform exceptionally basic functions. CPU functions include add, subtract, multiply and divide. The sequences and general structure of a processing unit and motherboard often change between computers, and hence machine code is often specific to a family of processors.

The reason behind only two values being used is as a result of the physical electronics behind everything, and most electronic components only being able to read whether there is or is no signal.

All programs and code must be broken down to machine code (aka binary) through each generation below for them to execute. As such, machine code is considered the lowest level language.

## Second Gen

Second generation languages often known as assembly languages are the first bridge between unreadable pulses and human communication. Some commands available in the language *Assembly* include `MOV`, `INC`, `AND`, `OR` and more. Although this makes computers infinitely easier to control, the programmer must still handle many operations themselves such as file management, memory management and the dressing of registers.

Second generation languages are still considered low-level languages as they are still hardware specific.

## Third Gen

These were the first of the high-level languages, and came around in the 1950s. They were used in early business and scientific apps with digestible syntaxes that could be learned and implemented more easily, allowing more people to become programmers. 3GLs introduced the concept of object-oriented programming, in which code did not execute sequentially but would instead jump between different sections of the code, or objects.

Some early third generations languages include FORTRAN, COBOL, BASIC, Pascal and the original C. Even Java, C++, C#, Python and JavaScript are often considered 3GLs.

In these languages, the computer begins to take care of the aforementioned duties of managing the computers resources, removing another layer of complexity.

#### **Fourth Gen**

Fourth generation languages were created to make programming again more accessible and increase development speed essentially by allowing for lazy programming (often done by removing the strict typing of variables and complicated syntax present in some 3GL's). They also more closely resemble natural human language, with some such as AppleScript being almost readable even to a non-programmer.

Some fourth generation languages include LabVIEW, MATLAB and SQL.

#### **Fifth Gen**

Fifth generation 'languages' do not comprise of written algorithms as present in all other generations, but simply run with a series of inputs and constraints. The purpose of this is to allow the computer to come up with its own solution without a specific set of instructions. This type of programming is most commonly known as machine learning or artificial intelligence.

In fifth generation programming there is no code that is directly involved in finding a solution, but code still must be written to guide and feed data to the black box that is the computers own solution. Common languages associated with machine learning include Python, R, and Java.

---

#### **References**

- Leach, J. (2013, November 13). The 5 Generations of Programming Languages. Retrieved June 29, 2018, from <https://cowputing.wordpress.com/2013/11/13/the-5-generations-of-programming-languages/>
- Hagerty, K. (2009, December 23). 5 Generations of Computer Languages. Retrieved June 29, 2018, from <https://www.mindmeister.com/37794301/5-generations-of-computer-languages>