A Brief History of Programming Languages

A *programming language* is a set of words, codes, and symbols that allow a programmer to give instructions to the computer. Many programming languages exist, each with their own rules, or syntax, for writing these instructions.

Programming languages can be classified as low-level and high-level languages. *Low-level programming languages* include machine language and assembly language. Machine language, which is referred to as a first generation programming language, can be used to communicate directly with the computer. However, it is difficult to program in machine language because the language consists of 0s and 1s to represent the status of a switch (0 for off and 1 for on). Assembly language uses the same instructions and structure as machine language but the programmer is able to use meaningful names or abbreviations instead of numbers. Assembly language is referred to as a second generation programming language.

High-level programming languages, which are often referred to as third generation programming languages (3GL), were first developed in the late 1950s. High-level programming languages have English-like instructions and are easier to use than machine language. High-level programming languages include Fortran, C, BASIC, COBOL, and Pascal. In order for the computer to understand a program written in a high-level language, programmers convert the source code into machine language using a compiler or an interpreter. A *compiler* is a program that converts an entire program into machine code before the program is executed. An *interpreter* translates and executes an instruction before moving on to the next instruction in the program.

In the 1980s, *object-oriented programming* (OOP) evolved out of the need to better develop complex programs in a systematic, organized approach. The OOP approach allows programmers to create modules that can be used over and over again in a variety of programs. These modules contain code called classes, which group related data and actions. Properly designed classes encapsulate data to hide the implementation details, are versatile enough to be extended through inheritance, and give the programmer options through polymorphism. Object-oriented languages include Java, C++ and Visual Basic.

Grace Murray Hopper 1906 – 1992

The first compiler, A-0, was created in 1951 by Grace Murray Hopper, a Commodore in the Navy at the time. She is also credited with developing COBOL with the United States Department of Defense in 1959. She later used the term "debug" when removing a moth that had flown into the circuitry of a Mark II computer.

Credits

In 1957, John Backus and a team of researchers developed Fortran.

In the 1960s, John Kemeny and Thomas Kurtz developed BASIC at Dartmouth University.

In the late 1970s, the United States Department of Defense developed Ada, a high-level language that supports real-time applications. Also in the 1970s, the C programming language was created by Dennis Ritchie at Bell Laboratories.

Fourth and Fifth Generation Languages

Fourth generation languages (4GL), such as SQL, have higher English-like instructions than most high-level languages and are typically used to access databases. Fifth generation languages are used for artificial intelligence.