**Changes in programming language**  **programming language** is a [formal language](https://en.wikipedia.org/wiki/Formal_language) which comprises a [set of instructions](https://en.wikipedia.org/wiki/Instruction_set) used to produce various kinds of [output](https://en.wikipedia.org/wiki/Input/output). Programming languages are used to create [programs](https://en.wikipedia.org/wiki/Program_(machine)) that implement specific [algorithms](https://en.wikipedia.org/wiki/Algorithm).

**Early developments**[

Very early computers, such as [Colossus](https://en.wikipedia.org/wiki/Colossus_computer), were programmed without the help of a [stored program](https://en.wikipedia.org/wiki/Stored_program), by modifying their circuitry or setting banks of physical controls.

Slightly later, programs could be written in [machine language](https://en.wikipedia.org/wiki/Machine_language), where the programmer writes each instruction in a numeric form the hardware can execute directly. For example, the instruction to add the value in two memory location might consist of 3 numbers: a "opcode" that selects the "add" operation, and two memory locations. The programs, in decimal or binary form, were read in from [punched cards](https://en.wikipedia.org/wiki/Punched_card) or [magnetic tape](https://en.wikipedia.org/wiki/Magnetic_tape_data_storage) or toggled in on switches on the [front panel](https://en.wikipedia.org/wiki/Front_panel) of the computer. Machine languages were later termed [*first-generation programming languages*](https://en.wikipedia.org/wiki/First-generation_programming_language) (1GL).

The next step was development of so-called [*second-generation programming languages*](https://en.wikipedia.org/wiki/Second-generation_programming_language) (2GL) or [assembly languages](https://en.wikipedia.org/wiki/Assembly_language), which were still closely tied to the [instruction set architecture](https://en.wikipedia.org/wiki/Instruction_set_architecture) of the specific computer. These served to make the program much more human-readable and relieved the programmer of tedious and error-prone address calculations.

The first [*high-level programming languages*](https://en.wikipedia.org/wiki/High-level_programming_language), or [*third-generation programming languages*](https://en.wikipedia.org/wiki/Third-generation_programming_language) (3GL), were written in the 1950s. An early high-level programming language to be designed for a computer was  [Plankalkül](https://en.wikipedia.org/wiki/Plankalk%C3%BCl" \o "Plankalkül), developed for the German [Z3](https://en.wikipedia.org/wiki/Z3_(computer)) by [Konrad Zuse](https://en.wikipedia.org/wiki/Konrad_Zuse" \o "Konrad Zuse) between 1943 and 1945. However, it was not implemented until 1998 and 2000.

**Refinement**

The increased use of high-level languages introduced a requirement for [*low-level programming languages*](https://en.wikipedia.org/wiki/Low-level_programming_language) or [*system programming languages*](https://en.wikipedia.org/wiki/System_programming_language). These languages, to varying degrees, provide facilities between assembly languages and high-level languages and can be used to perform tasks which require direct access to hardware facilities but still provide higher-level control structures and error-checking.

The period from the 1960s to the late 1970s brought the development of the major language paradigms now in use:

* [APL](https://en.wikipedia.org/wiki/APL_(programming_language)) introduced [*array programming*](https://en.wikipedia.org/wiki/Array_programming) and influenced [functional programming](https://en.wikipedia.org/wiki/Functional_programming).
* [ALGOL](https://en.wikipedia.org/wiki/ALGOL) refined both *structured procedural programming* and the discipline of [language specification](https://en.wikipedia.org/wiki/Programming_language_specification); the "Revised Report on the Algorithmic Language [ALGOL 60](https://en.wikipedia.org/wiki/ALGOL_60)" became a model for how later language specifications were written.
* [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)), implemented in 1958, was the first dynamically typed [*functional programming*](https://en.wikipedia.org/wiki/Functional_programming) language
* In the 1960s, [Simula](https://en.wikipedia.org/wiki/Simula" \o "Simula) was the first language designed to support [*object-oriented programming*](https://en.wikipedia.org/wiki/Object-oriented_programming); in the mid-1970s, [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk) followed with the first "purely" object-oriented language.
* [C](https://en.wikipedia.org/wiki/C_(programming_language)) was developed between 1969 and 1973 as a system programming language for the [Unix](https://en.wikipedia.org/wiki/Unix) operating system and remains popular.[[38]](https://en.wikipedia.org/wiki/Programming_language#cite_note-38)
* [Prolog](https://en.wikipedia.org/wiki/Prolog), designed in 1972, was the first [*logic programming*](https://en.wikipedia.org/wiki/Logic_programming) language.
* In 1978, [ML](https://en.wikipedia.org/wiki/ML_(programming_language)) built a polymorphic type system on top of [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)), pioneering [*statically typed*](https://en.wikipedia.org/wiki/Type_system)[*functional programming*](https://en.wikipedia.org/wiki/Functional_programming) languages.

Each of these languages spawned descendants, and most modern programming languages count at least one of them in their ancestry.

The 1960s and 1970s also saw considerable debate over the merits of [*structured programming*](https://en.wikipedia.org/wiki/Structured_programming), and whether programming languages should be designed to support it.[[39]](https://en.wikipedia.org/wiki/Programming_language#cite_note-39) [Edsger Dijkstra](https://en.wikipedia.org/wiki/Edsger_W._Dijkstra" \o "Edsger W. Dijkstra), in a famous 1968 letter published in the [Communications of the ACM](https://en.wikipedia.org/wiki/Communications_of_the_ACM), argued that [GOTO](https://en.wikipedia.org/wiki/Goto) statements should be eliminated from all "higher level" programming languages.