

Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. There are many approaches to the Software development process. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Integrated development environments (IDEs) aim to integrate all such help. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. It is very difficult to determine what are the most popular modern programming languages. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Techniques like Code refactoring can enhance readability. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. It is very difficult to determine what are the most popular modern programming languages. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.