

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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. However, readability is more than just programming style. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Ideally, the programming language best suited for the task at hand will be selected. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Ideally, the programming language best suited for the task at hand will be selected. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. 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. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Ideally, the programming language best suited for the task at hand will be selected. Normally the first step in debugging is to attempt to reproduce the problem. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Techniques like Code refactoring can enhance readability. However, readability is more than just programming style. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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).