

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. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. 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.

Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Many applications use a mix of several languages in their construction and use. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. Ideally, the programming language best suited for the task at hand will be selected. Programmable devices have existed for centuries. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. There are many approaches to the Software development process. 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). 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Scripting and breakpointing is also part of this process.