

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. Programs were mostly entered using punched cards or paper tape. 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. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. There are many approaches to the Software development process. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Programmable devices have existed for centuries. Programmable devices have existed for centuries. 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. 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. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. 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). Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. Integrated development environments (IDEs) aim to integrate all such help. Scripting and breakpointing is also part of this process. Programmable devices have existed for centuries. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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.