

Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. Programs were mostly entered using punched cards or paper tape. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. However, readability is more than just programming style. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Programming languages are essential for software development. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. There exist a lot of different approaches for each of those tasks. Scripting and breakpointing is also part of this process. 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. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Following a consistent programming style often helps readability. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. One approach popular for requirements analysis is Use Case analysis. Integrated development environments (IDEs) aim to integrate all such help. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Many applications use a mix of several languages in their construction and use. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. There are many approaches to the Software development process. Ideally, the programming language best suited for the task at hand will be selected.