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. Programs were mostly entered using punched cards or paper tape. 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. 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. 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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Programmable devices have existed for centuries. Programmable devices have existed for centuries. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. One approach popular for requirements analysis is Use Case analysis. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Different programming languages support different styles of programming (called programming paradigms). Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. Following a consistent programming style often helps readability. 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. Code-breaking algorithms have also existed for centuries.