

In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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). However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. 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). Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. One approach popular for requirements analysis is Use Case analysis. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Different programming languages support different styles of programming (called programming paradigms). 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.