It is usually easier to code in "high-level" languages than in "low-level" ones. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Many applications use a mix of several languages in their construction and use. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Different programming languages support different styles of programming (called programming paradigms). 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. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. 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. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. It is usually easier to code in "high-level" languages than in "low-level" ones. 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 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Integrated development environments (IDEs) aim to integrate all such help. However, readability is more than just programming style. 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. It is very difficult to determine what are the most popular modern programming languages.