Integrated development environments (IDEs) aim to integrate all such help. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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). Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. 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. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. There exist a lot of different approaches for each of those tasks. Use of a static code analysis tool can help detect some possible problems. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. 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. 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. Normally the first step in debugging is to attempt to reproduce the problem. 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.