

However, readability is more than just programming style. 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. Different programming languages support different styles of programming (called programming paradigms). Use of a static code analysis tool can help detect some possible problems. Techniques like Code refactoring can enhance 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. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. 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. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. 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. 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). Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. However, readability is more than just programming style. 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. Different programming languages support different styles of programming (called programming paradigms). 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 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Many applications use a mix of several languages in their construction and use.