

In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. There are many approaches to the Software development process. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Programming languages are essential for software development. 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. Programs were mostly entered using punched cards or paper tape. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. There are many approaches to the Software development process. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. 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.