

Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Different programming languages support different styles of programming (called programming paradigms). They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. 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. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Integrated development environments (IDEs) aim to integrate all such help. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. 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. 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. Computer programmers are those who write computer software. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. 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. 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. One approach popular for requirements analysis is Use Case analysis.