

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. Code-breaking algorithms have also existed for centuries. Techniques like Code refactoring can enhance readability. 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). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Computer programmers are those who write computer software. There exist a lot of different approaches for each of those tasks. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. Techniques like Code refactoring can enhance readability. Ideally, the programming language best suited for the task at hand will be selected. 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. Programs were mostly entered using punched cards or paper tape. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Many applications use a mix of several languages in their construction and use. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. It is usually easier to code in "high-level" languages than in "low-level" ones. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.