

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. 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. Techniques like Code refactoring can enhance readability. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Techniques like Code refactoring can enhance readability. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Computer programmers are those who write computer software. Different programming languages support different styles of programming (called programming paradigms). Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Scripting and breakpointing is also part of this process. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.