

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. 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. 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. Techniques like Code refactoring can enhance readability. Use of a static code analysis tool can help detect some possible problems. One approach popular for requirements analysis is Use Case analysis. Code-breaking algorithms have also existed for centuries. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Following a consistent programming style often helps readability. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Many applications use a mix of several languages in their construction and use. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Following a consistent programming style often helps readability. 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. 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.