

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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Following a consistent programming style often helps readability. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. One approach popular for requirements analysis is Use Case analysis. However, readability is more than just programming style. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. 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. However, readability is more than just programming style. Use of a static code analysis tool can help detect some possible problems. 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. 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. 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. Code-breaking algorithms have also existed for centuries. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. It is usually easier to code in "high-level" languages than in "low-level" ones.