

Integrated development environments (IDEs) aim to integrate all such help. 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 first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.

Code-breaking algorithms have also existed for centuries. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Programs were mostly entered using punched cards or paper tape. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. 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. Code-breaking algorithms have also existed for centuries. Unreadable code often leads to bugs, inefficiencies, and duplicated code. However, readability is more than just programming style. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Programs were mostly entered using punched cards or paper tape.