

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 very difficult to determine what are the most popular modern programming languages. Techniques like Code refactoring can enhance readability. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Following a consistent programming style often helps readability. Computer programmers are those who write computer software. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. 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. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Following a consistent programming style often helps readability. Techniques like Code refactoring can enhance readability. 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. 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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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.