

It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Code-breaking algorithms have also existed for centuries. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Following a consistent programming style often helps readability. 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. 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. Ideally, the programming language best suited for the task at hand will be selected. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. 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. 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. There are many approaches to the Software development process. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Whatever the approach to development may be, the final program must satisfy some fundamental properties.