

It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Ideally, the programming language best suited for the task at hand will be selected. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. It is very difficult to determine what are the most popular modern programming languages. 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. Use of a static code analysis tool can help detect some possible problems. Programming languages are essential for software development. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. Programmable devices have existed for centuries. 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. 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. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. It is usually easier to code in "high-level" languages than in "low-level" ones. However, readability is more than just programming style. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Techniques like Code refactoring can enhance readability.