

Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Following a consistent programming style often helps readability. Many applications use a mix of several languages in their construction and use. Use of a static code analysis tool can help detect some possible problems. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. 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. 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. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. There exist a lot of different approaches for each of those tasks. 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. Ideally, the programming language best suited for the task at hand will be selected. Normally the first step in debugging is to attempt to reproduce the problem. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly 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. Techniques like Code refactoring can enhance readability. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries.