

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. It is usually easier to code in "high-level" languages than in "low-level" ones. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. 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. Scripting and breakpointing is also part of this process. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Normally the first step in debugging is to attempt to reproduce the problem. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. One approach popular for requirements analysis is Use Case analysis. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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.