

However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. There exist a lot of different approaches for each of those tasks. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Programming languages are essential for software development. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.

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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. There exist a lot of different approaches for each of those 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.