

Code-breaking algorithms have also existed for centuries. Different programming languages support different styles of programming (called programming paradigms). Whatever the approach to development may be, the final program must satisfy some fundamental properties. Following a consistent programming style often helps readability. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Scripting and breakpointing is also part of this process. 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. There exist a lot of different approaches for each of those tasks. Programs were mostly entered using punched cards or paper tape. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Use of a static code analysis tool can help detect some possible problems. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. However, readability is more than just programming style. 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. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.