

High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Programs were mostly entered using punched cards or paper tape. One approach popular for requirements analysis is Use Case analysis. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Different programming languages support different styles of programming (called programming paradigms). A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). It is very difficult to determine what are the most popular modern programming languages. It is usually easier to code in "high-level" languages than in "low-level" ones. There are many approaches to the Software development process. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Many applications use a mix of several languages in their construction and use. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. 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. Following a consistent programming style often helps readability. It is usually easier to code in "high-level" languages than in "low-level" ones. Integrated development environments (IDEs) aim to integrate all such help. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. There exist a lot of different approaches for each of those tasks. However, readability is more than just programming style. It is usually easier to code in "high-level" languages than in "low-level" ones. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.