

Integrated development environments (IDEs) aim to integrate all such help. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Computer programmers are those who write computer software. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Use of a static code analysis tool can help detect some possible problems. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. 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. Use of a static code analysis tool can help detect some possible problems. Integrated development environments (IDEs) aim to integrate all such help. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. Code-breaking algorithms have also existed for centuries. 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. 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. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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.