Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. It is very difficult to determine what are the most popular modern programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Programs were mostly entered using punched cards or paper tape. 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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. 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, readability is more than just programming style. 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. 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 Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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.