The Unified Modeling Language (UML) is a notation used for both the OOAD and 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. 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. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. 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. 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. Ideally, the programming language best suited for the task at hand will be selected. Computer programmers are those who write computer software. 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). 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. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. However, readability is more than just programming style. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. There exist a lot of different approaches for each of those tasks. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. Programming languages are essential for software development. Code-breaking algorithms have also existed for centuries. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. 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 first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. 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. Following a consistent programming style often helps readability. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.