

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. Programmable devices have existed for centuries. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. 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. There exist a lot of different approaches for each of those tasks. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Techniques like Code refactoring can enhance readability. 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. 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. 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. 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*. Ideally, the programming language best suited for the task at hand will be selected. Code-breaking algorithms have also existed for centuries. However, readability is more than just programming style. Following a consistent programming style often helps readability. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. 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.