Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. 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. Scripting and breakpointing is also part of this process. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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. One approach popular for requirements analysis is Use Case analysis. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Code-breaking algorithms have also existed for centuries. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Techniques like Code refactoring can enhance readability. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. 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). 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. Programs were mostly entered using punched cards or paper tape. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Programs were mostly entered using punched cards or paper tape.