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. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Following a consistent programming style often helps readability. 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). One approach popular for requirements analysis is Use Case analysis. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. Scripting and breakpointing is also part of this process. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. 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. 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. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Ideally, the programming language best suited for the task at hand will be selected. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.