He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Unreadable code often leads to bugs, inefficiencies, and duplicated code. One approach popular for requirements analysis is Use Case analysis. 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. Scripting and breakpointing is also part of this process. It is usually easier to code in "high-level" languages than in "low-level" ones. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It is very difficult to determine what are the most popular modern programming languages. 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. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. It is usually easier to code in "high-level" languages than in "low-level" ones. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Ideally, the programming language best suited for the task at hand will be selected. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.