Many applications use a mix of several languages in their construction and use. Scripting and breakpointing is also part of this process. 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. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Many applications use a mix of several languages in their construction and use. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Programmable devices have existed for centuries. 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. 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. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. 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. However, readability is more than just programming style. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. However, readability is more than just programming style. Many applications use a mix of several languages in their construction and use. Programming languages are essential for software development.