High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. Techniques like Code refactoring can enhance readability. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. 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. 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. Different programming languages support different styles of programming (called programming paradigms). 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). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. 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.