Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Computer programmers are those who write computer software. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. 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. There exist a lot of different approaches for each of those tasks. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. There exist a lot of different approaches for each of those tasks. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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.