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. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Unreadable code often leads to bugs, inefficiencies, and duplicated code. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Different programming languages support different styles of programming (called programming paradigms). Also, specific user environment and usage history can make it difficult to reproduce the problem. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Following a consistent programming style often helps readability. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Different programming languages support different styles of programming (called programming paradigms). In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. 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. Ideally, the programming language best suited for the task at hand will be selected. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Integrated development environments (IDEs) aim to integrate all such help.