Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. There are many approaches to the Software development process. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). 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. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Ideally, the programming language best suited for the task at hand will be selected. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Techniques like Code refactoring can enhance readability. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Different programming languages support different styles of programming (called programming paradigms). New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). It is very difficult to determine what are the most popular modern programming languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Programming languages are essential for software development. Also, specific user environment and usage history can make it difficult to reproduce the problem. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. It is usually easier to code in "high-level" languages than in "low-level" ones. Different programming languages support different styles of programming (called programming paradigms).