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. Integrated development environments (IDEs) aim to integrate all such help. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Programming languages are essential for software development. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). There are many approaches to the Software development process. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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 involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Many applications use a mix of several languages in their construction and use. It is usually easier to code in "high-level" languages than in "low-level" ones. Techniques like Code refactoring can enhance readability. Code-breaking algorithms have also existed for centuries. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. 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. 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. Following a consistent programming style often helps readability. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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 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.