It affects the aspects of quality above, including portability, usability and most importantly maintainability. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Also, specific user environment and usage history can make it difficult to reproduce the problem. Ideally, the programming language best suited for the task at hand will be selected. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Ideally, the programming language best suited for the task at hand will be selected. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. 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.