

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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. There are many approaches to the Software development process. Ideally, the programming language best suited for the task at hand will be selected. 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. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. 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 first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. It is usually easier to code in "high-level" languages than in "low-level" ones. Whatever the approach to development may be, the final program must satisfy some fundamental properties.