

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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Programming languages are essential for software development. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. One approach popular for requirements analysis is Use Case analysis. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Ideally, the programming language best suited for the task at hand will be selected. 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). 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, readability is more than just programming style. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Many applications use a mix of several languages in their construction and use. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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). Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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.