

In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. One approach popular for requirements analysis is Use Case analysis. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Techniques like Code refactoring can enhance readability. Following a consistent programming style often helps readability. Unreadable code often leads to bugs, inefficiencies, and duplicated code. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. There exist a lot of different approaches for each of those tasks. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Integrated development environments (IDEs) aim to integrate all such help. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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). Computer programmers are those who write computer software. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Unreadable code often leads to bugs, inefficiencies, and duplicated code. However, readability is more than just programming style.