

Computer programmers are those who write computer software. Code-breaking algorithms have also existed for centuries. Techniques like Code refactoring can enhance readability. It is usually easier to code in "high-level" languages than in "low-level" ones. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. However, readability is more than just programming style. 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. Integrated development environments (IDEs) aim to integrate all such help. It is usually easier to code in "high-level" languages than in "low-level" ones. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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). Also, specific user environment and usage history can make it difficult to reproduce the problem. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. Integrated development environments (IDEs) aim to integrate all such help. 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. Following a consistent programming style often helps readability. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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.