

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. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. 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. Techniques like Code refactoring can enhance readability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Scripting and breakpointing is also part of this process. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. It affects the aspects of quality above, including portability, usability and most importantly maintainability. There are many approaches to the Software development process. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Use of a static code analysis tool can help detect some possible problems. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Programs were mostly entered using punched cards or paper tape. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. There exist a lot of different approaches for each of those tasks. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Scripting and breakpointing is also part of this process. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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.