

After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. However, readability is more than just programming style. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Techniques like Code refactoring can enhance readability. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Also, specific user environment and usage history can make it difficult to reproduce the problem. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. It is usually easier to code in "high-level" languages than in "low-level" ones. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Unreadable code often leads to bugs, inefficiencies, and duplicated code. Use of a static code analysis tool can help detect some possible problems. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Also, specific user environment and usage history can make it difficult to reproduce the problem.