

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. 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. One approach popular for requirements analysis is Use Case analysis. Normally the first step in debugging is to attempt to reproduce the problem. 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. Different programming languages support different styles of programming (called programming paradigms). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. Integrated development environments (IDEs) aim to integrate all such help. Following a consistent programming style often helps readability. One approach popular for requirements analysis is Use Case analysis. Scripting and breakpointing is also part of this process. There are many approaches to the Software development process. 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. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. 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. 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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).