

Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design 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 text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Use of a static code analysis tool can help detect some possible problems. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Programming languages are essential for software development. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. Techniques like Code refactoring can enhance readability. Computer programmers are those who write computer software. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Techniques like Code refactoring can enhance readability. It affects the aspects of quality above, including portability, usability and most importantly maintainability.