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..  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 Programmable devices have existed for centuries.  
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.  
  
 Different programming languages support different styles of programming (called programming paradigms).  
Ideally, the programming language best suited for the task at hand will be selected.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.