By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Techniques like Code refactoring can enhance readability.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
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.  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Ideally, the programming language best suited for the task at hand will be selected.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.