However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
However, readability is more than just programming style.  
However, readability is more than just programming style.  
There exist a lot of different approaches for each of those tasks.  
 Programmable devices have existed for centuries.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
  
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.