There exist a lot of different approaches for each of those tasks.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
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.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 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.  
 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.