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..  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 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.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Techniques like Code refactoring can enhance readability.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
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.  
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.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
Integrated development environments (IDEs) aim to integrate all such help.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.