Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards..  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 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.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
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.  
 Following a consistent programming style often helps readability.  
 Different programming languages support different styles of programming (called programming paradigms).