In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages..  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
However, readability is more than just programming style.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
 Code-breaking algorithms have also existed for centuries.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.