Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
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.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
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 affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
However, readability is more than just programming style.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Code-breaking algorithms have also existed for centuries.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.