Compiling takes the source code from a low-level programming language and converts it into machine code.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
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.  
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.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.