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.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Programming languages are essential for software development.  
However, readability is more than just programming style.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
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.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
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.  
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.  
 Following a consistent programming style often helps readability.