Techniques like Code refactoring can enhance readability.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
There are many approaches to the Software development process.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
However, readability is more than just programming style.  
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.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Scripting and breakpointing is also part of this process.  
This is interpreted into machine code.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Ideally, the programming language best suited for the task at hand will be selected.