Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation..  
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.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
However, readability is more than just programming style.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
There exist a lot of different approaches for each of those tasks.  
Scripting and breakpointing is also part of this process.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Techniques like Code refactoring can enhance readability.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
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.  
 Computer programmers are those who write computer software.  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.  
It is usually easier to code in "high-level" languages than in "low-level" ones.