The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
To produce machine code, the source code must either be compiled or transpiled.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
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).  
Integrated development environments (IDEs) aim to integrate all such help.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 It is very difficult to determine what are the most popular modern programming languages.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.