To produce machine code, the source code must either be compiled or transpiled.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
There exist a lot of different approaches for each of those tasks.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
One approach popular for requirements analysis is Use Case analysis.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
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.  
Integrated development environments (IDEs) aim to integrate all such help.