The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging)..  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Different programming languages support different styles of programming (called programming paradigms).  
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.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
There exist a lot of different approaches for each of those tasks.  
 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.  
  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
There are many approaches to the Software development process.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).