One approach popular for requirements analysis is Use Case analysis.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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 factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
One approach popular for requirements analysis is Use Case analysis.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 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.  
 Different programming languages support different styles of programming (called programming paradigms).  
Many applications use a mix of several languages in their construction and use.