Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages..  
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.  
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.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Integrated development environments (IDEs) aim to integrate all such help.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
One approach popular for requirements analysis is Use Case analysis.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
 Different programming languages support different styles of programming (called programming paradigms).  
Techniques like Code refactoring can enhance readability.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
There are many approaches to the Software development process.  
  
Normally the first step in debugging is to attempt to reproduce the problem.