It is usually easier to code in "high-level" languages than in "low-level" ones..  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 Code-breaking algorithms have also existed for centuries.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
Many applications use a mix of several languages in their construction and use.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.