It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
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.  
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.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
This is interpreted into machine code.  
There are many approaches to the Software development process.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.