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..  
 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.  
 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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
Integrated development environments (IDEs) aim to integrate all such help.  
Use of a static code analysis tool can help detect some possible problems.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
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.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There exist a lot of different approaches for each of those tasks.