These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics..  
 Programmable devices have existed for centuries.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Scripting and breakpointing is also part of this process.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
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.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
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.