Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications..  
 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.  
Many applications use a mix of several languages in their construction and use.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
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.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
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.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Use of a static code analysis tool can help detect some possible problems.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.