It is usually easier to code in "high-level" languages than in "low-level" ones..  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
One approach popular for requirements analysis is Use Case analysis.  
Scripting and breakpointing is also part of this process.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
 It is very difficult to determine what are the most popular modern programming languages.  
However, readability is more than just programming style.  
 Different programming languages support different styles of programming (called programming paradigms).  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Programs were mostly entered using punched cards or paper tape.  
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.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.