Many applications use a mix of several languages in their construction and use.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Normally the first step in debugging is to attempt to reproduce the problem.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
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.  
 Different programming languages support different styles of programming (called programming paradigms).  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
There exist a lot of different approaches for each of those tasks.  
Use of a static code analysis tool can help detect some possible problems.