Unreadable code often leads to bugs, inefficiencies, and duplicated code..  
  
 Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
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.  
 Code-breaking algorithms have also existed for centuries.  
 Different programming languages support different styles of programming (called programming paradigms).  
Use of a static code analysis tool can help detect some possible problems.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
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.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.