Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit..  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
Ideally, the programming language best suited for the task at hand will be selected.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
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.  
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.  
 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.  
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.