A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
To produce machine code, the source code must either be compiled or transpiled.  
Integrated development environments (IDEs) aim to integrate all such help.  
To produce machine code, the source code must either be compiled or transpiled.  
 Following a consistent programming style often helps readability.  
 Computer programmers are those who write computer software.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 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).  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Scripting and breakpointing is also part of this process.