Use of a static code analysis tool can help detect some possible problems.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Ideally, the programming language best suited for the task at hand will be selected.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
Many applications use a mix of several languages in their construction and use.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, readability is more than just programming style.  
Scripting and breakpointing is also part of this process.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
This is interpreted into machine code.  
 Different programming languages support different styles of programming (called programming paradigms).  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.