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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
However, readability is more than just programming style.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
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.  
However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
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.