This can be a non-trivial task, for example as with parallel processes or some unusual software bugs..  
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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
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.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Techniques like Code refactoring can enhance readability.  
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.  
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.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.