This can be a non-trivial task, for example as with parallel processes or some unusual software bugs..  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Different programming languages support different styles of programming (called programming paradigms).  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 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.  
Use of a static code analysis tool can help detect some possible problems.  
 It is very difficult to determine what are the most popular modern programming languages.  
Techniques like Code refactoring can enhance readability.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
Scripting and breakpointing is also part of this process.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.