Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process..  
Techniques like Code refactoring can enhance readability.  
Scripting and breakpointing is also part of this process.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
There exist a lot of different approaches for each of those tasks.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
 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.  
 It is very difficult to determine what are the most popular modern programming languages.