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.  
Use of a static code analysis tool can help detect some possible problems.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
There exist a lot of different approaches for each of those tasks.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Programming languages are essential for software development.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.