Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
There exist a lot of different approaches for each of those tasks.  
To produce machine code, the source code must either be compiled or transpiled.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Code-breaking algorithms have also existed for centuries.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
This is interpreted into machine code.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.  
Techniques like Code refactoring can enhance readability.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
 Different programming languages support different styles of programming (called programming paradigms).  
Integrated development environments (IDEs) aim to integrate all such help.