This is interpreted into machine code.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
There exist a lot of different approaches for each of those tasks.  
This is interpreted into machine code.  
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.  
This is interpreted into machine code.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
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.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
However, readability is more than just programming style.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.