This is interpreted into machine code.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
However, readability is more than just programming style.  
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.  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
Techniques like Code refactoring can enhance readability.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
There exist a lot of different approaches for each of those tasks.  
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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.