Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Normally the first step in debugging is to attempt to reproduce the problem.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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).  
Scripting and breakpointing is also part of this process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Ideally, the programming language best suited for the task at hand will be selected.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Techniques like Code refactoring can enhance readability.