Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Ideally, the programming language best suited for the task at hand will be selected.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
There are many approaches to the Software development process.  
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).  
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.  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.