There are many approaches to the Software development process.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
One approach popular for requirements analysis is Use Case analysis.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.