Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
However, readability is more than just programming style.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
Many applications use a mix of several languages in their construction and use.  
There are many approaches to the Software development process.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
Ideally, the programming language best suited for the task at hand will be selected.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
There exist a lot of different approaches for each of those tasks.