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).  
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.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
There exist a lot of different approaches for each of those tasks.  
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.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Integrated development environments (IDEs) aim to integrate all such help.  
This is interpreted into machine code.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Different programming languages support different styles of programming (called programming paradigms).  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.