Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
One approach popular for requirements analysis is Use Case analysis.  
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).  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 Following a consistent programming style often helps readability.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.