Ideally, the programming language best suited for the task at hand will be selected.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
This is interpreted into machine code.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
Many applications use a mix of several languages in their construction and use.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Use of a static code analysis tool can help detect some possible problems.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
To produce machine code, the source code must either be compiled or transpiled.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
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.  
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.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.