Ideally, the programming language best suited for the task at hand will be selected..  
 Different programming languages support different styles of programming (called programming paradigms).  
One approach popular for requirements analysis is Use Case analysis.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
Techniques like Code refactoring can enhance readability.  
Integrated development environments (IDEs) aim to integrate all such help.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
Use of a static code analysis tool can help detect some possible problems.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.