Relatedly, software engineering combines engineering techniques and principles with software development.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Normally the first step in debugging is to attempt to reproduce the problem.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
This is interpreted into machine code.  
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 very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Following a consistent programming style often helps readability.  
One approach popular for requirements analysis is Use Case analysis.