High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware..  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
Scripting and breakpointing is also part of this process.  
Many applications use a mix of several languages in their construction and use.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).