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..  
Techniques like Code refactoring can enhance readability.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Programs were mostly entered using punched cards or paper tape.  
 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.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Ideally, the programming language best suited for the task at hand will be selected.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
There are many approaches to the Software development process.  
There exist a lot of different approaches for each of those tasks.