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..  
One approach popular for requirements analysis is Use Case analysis.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
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.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
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.  
Many applications use a mix of several languages in their construction and use.  
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.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.