High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware..  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
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.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
One approach popular for requirements analysis is Use Case analysis.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Code-breaking algorithms have also existed for centuries.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
 Different programming languages support different styles of programming (called programming paradigms).  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).