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..  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
 Different programming languages support different styles of programming (called programming paradigms).  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Code-breaking algorithms have also existed for centuries.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
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.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.