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..  
Use of a static code analysis tool can help detect some possible problems.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
 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.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Different programming languages support different styles of programming (called programming paradigms).  
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.  
  
 Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Ideally, the programming language best suited for the task at hand will be selected.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.