The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging)..  
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 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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
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.  
  
 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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
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.  
 Programs were mostly entered using punched cards or paper tape.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Different programming languages support different styles of programming (called programming paradigms).  
It is usually easier to code in "high-level" languages than in "low-level" ones.