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..  
 Following a consistent programming style often helps readability.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Integrated development environments (IDEs) aim to integrate all such help.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
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.  
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.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
There exist a lot of different approaches for each of those tasks.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.