Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
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.  
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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.