He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm..  
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.  
 Programmable devices have existed for centuries.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
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.  
 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).  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.