Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
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.  
There exist a lot of different approaches for each of those tasks.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
Programming languages are essential for software development.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.