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..  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Code-breaking algorithms have also existed for centuries.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
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.  
 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.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.