Techniques like Code refactoring can enhance 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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
To produce machine code, the source code must either be compiled or transpiled.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Many applications use a mix of several languages in their construction and use.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
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.  
This is interpreted into machine code.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.