To produce machine code, the source code must either be compiled or transpiled.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
There exist a lot of different approaches for each of those tasks.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Many applications use a mix of several languages in their construction and use.  
Programming languages are essential for software development.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Programs were mostly entered using punched cards or paper tape.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
 Following a consistent programming style often helps readability.  
It is usually easier to code in "high-level" languages than in "low-level" ones.