Unreadable code often leads to bugs, inefficiencies, and duplicated code..  
 Programmable devices have existed for centuries.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
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.  
Scripting and breakpointing is also part of this process.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
There exist a lot of different approaches for each of those tasks.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Programming languages are essential for software development.