Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit..  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
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.  
 Following a consistent programming style often helps readability.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 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.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
There exist a lot of different approaches for each of those tasks.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
Scripting and breakpointing is also part of this process.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.