Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Scripting and breakpointing is also part of this process.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
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.  
Use of a static code analysis tool can help detect some possible problems.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
There exist a lot of different approaches for each of those tasks.  
Integrated development environments (IDEs) aim to integrate all such help.  
Techniques like Code refactoring can enhance readability.  
This is interpreted into machine code.  
 Programs were mostly entered using punched cards or paper tape.  
There exist a lot of different approaches for each of those tasks.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.