Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Ideally, the programming language best suited for the task at hand will be selected.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Normally the first step in debugging is to attempt to reproduce the problem.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
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.  
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.  
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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
 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.  
 It is very difficult to determine what are the most popular modern programming languages.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.