There are many approaches to the Software development process.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
To produce machine code, the source code must either be compiled or transpiled.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Integrated development environments (IDEs) aim to integrate all such help.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
 Following a consistent programming style often helps readability.  
There exist a lot of different approaches for each of those tasks.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Scripting and breakpointing is also part of this process.