This is interpreted into machine code.  
There are many approaches to the Software development process.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
  
 Programs were mostly entered using punched cards or paper tape.  
 Following a consistent programming style often helps readability.  
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.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Ideally, the programming language best suited for the task at hand will be selected.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.