Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Many applications use a mix of several languages in their construction and use.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There are many approaches to the Software development process.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
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.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
This is interpreted into machine code.  
Techniques like Code refactoring can enhance readability.