It is very difficult to determine what are the most popular modern programming languages..  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
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.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Programs were mostly entered using punched cards or paper tape.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
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.  
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.