Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute..  
Ideally, the programming language best suited for the task at hand will be selected.  
 It is very difficult to determine what are the most popular modern programming languages.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
However, readability is more than just programming style.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
There exist a lot of different approaches for each of those tasks.  
Many applications use a mix of several languages in their construction and use.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
 Code-breaking algorithms have also existed for centuries.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.