Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display..  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
  
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.  
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.  
 It is very difficult to determine what are the most popular modern programming languages.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.