Different programming languages support different styles of programming (called programming paradigms)..  
Programming languages are essential for software development.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
 Following a consistent programming style often helps readability.  
However, readability is more than just programming style.  
 Programmable devices have existed for centuries.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
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.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.