There are many approaches to the Software development process..  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
 Different programming languages support different styles of programming (called programming paradigms).  
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).  
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.  
 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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
Many applications use a mix of several languages in their construction and use.