Whatever the approach to development may be, the final program must satisfy some fundamental properties..  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
There are many approaches to the Software development process.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 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.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.