Integrated development environments (IDEs) aim to integrate all such help..  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
 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.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Techniques like Code refactoring can enhance readability.  
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.  
Many applications use a mix of several languages in their construction and use.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.