Integrated development environments (IDEs) aim to integrate all such help.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
Integrated development environments (IDEs) aim to integrate all such help.  
Techniques like Code refactoring can enhance readability.  
However, readability is more than just programming style.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Programmable devices have existed for centuries.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
 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.