Compiling takes the source code from a low-level programming language and converts it into machine code.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
There exist a lot of different approaches for each of those tasks.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Integrated development environments (IDEs) aim to integrate all such help.  
Scripting and breakpointing is also part of this process.  
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.  
However, readability is more than just programming style.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 Code-breaking algorithms have also existed for centuries.  
 Following a consistent programming style often helps readability.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.