There exist a lot of different approaches for each of those tasks.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
There are many approaches to the Software development process.  
To produce machine code, the source code must either be compiled or transpiled.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
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).  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Following a consistent programming style often helps readability.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
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.  
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.