There exist a lot of different approaches for each of those tasks.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Many applications use a mix of several languages in their construction and use.  
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.  
Scripting and breakpointing is also part of this process.  
Normally the first step in debugging is to attempt to reproduce the problem.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
To produce machine code, the source code must either be compiled or transpiled.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Ideally, the programming language best suited for the task at hand will be selected.  
However, readability is more than just programming style.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.