Ideally, the programming language best suited for the task at hand will be selected.  
However, readability is more than just programming style.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Techniques like Code refactoring can enhance readability.  
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.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
However, readability is more than just programming style.  
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.  
Many applications use a mix of several languages in their construction and use.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.