To produce machine code, the source code must either be compiled or transpiled.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
There are many approaches to the Software development process.  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Many applications use a mix of several languages in their construction and use.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
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).  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Use of a static code analysis tool can help detect some possible problems.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.