Ideally, the programming language best suited for the task at hand will be selected.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
There exist a lot of different approaches for each of those tasks.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.