Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Ideally, the programming language best suited for the task at hand will be selected.  
To produce machine code, the source code must either be compiled or transpiled.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
Use of a static code analysis tool can help detect some possible problems.  
Scripting and breakpointing is also part of this process.  
Integrated development environments (IDEs) aim to integrate all such help.  
Techniques like Code refactoring can enhance readability.  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
  
 Computer programming is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building executable computer programs.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.