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.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Use of a static code analysis tool can help detect some possible problems.  
To produce machine code, the source code must either be compiled or transpiled.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
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).  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.