Normally the first step in debugging is to attempt to reproduce the problem..  
Programming languages are essential for software development.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
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.  
 Following a consistent programming style often helps readability.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
There exist a lot of different approaches for each of those tasks.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
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.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.