Use of a static code analysis tool can help detect some possible problems..  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Many applications use a mix of several languages in their construction and use.  
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.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Different programming languages support different styles of programming (called programming paradigms).  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.