Use of a static code analysis tool can help detect some possible problems.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There exist a lot of different approaches for each of those tasks.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
 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.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Normally the first step in debugging is to attempt to reproduce the problem.  
  
 Computer programming is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building executable computer programs.  
Use of a static code analysis tool can help detect some possible problems.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.