FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
There are many approaches to the Software development process.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Scripting and breakpointing is also part of this process.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
This is interpreted into machine code.