A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
Techniques like Code refactoring can enhance readability.  
Ideally, the programming language best suited for the task at hand will be selected.  
However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
One approach popular for requirements analysis is Use Case analysis.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
Normally the first step in debugging is to attempt to reproduce the problem.  
One approach popular for requirements analysis is Use Case analysis.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.