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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
To produce machine code, the source code must either be compiled or transpiled.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
Use of a static code analysis tool can help detect some possible problems.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Use of a static code analysis tool can help detect some possible problems.  
Programming languages are essential for software development.  
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.  
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.  
This is interpreted into machine code.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.