To produce machine code, the source code must either be compiled or transpiled.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Scripting and breakpointing is also part of this process.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Programming languages are essential for software development.  
One approach popular for requirements analysis is Use Case analysis.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
Programming languages are essential for software development.  
Use of a static code analysis tool can help detect some possible problems.  
Programming languages are essential for software development.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).