Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
This is interpreted into machine code.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
There are many approaches to the Software development process.  
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.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Ideally, the programming language best suited for the task at hand will be selected.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
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.  
Programming languages are essential for software development.  
To produce machine code, the source code must either be compiled or transpiled.