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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
This is interpreted into machine code.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
Techniques like Code refactoring can enhance readability.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
Programming languages are essential for software development.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
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.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
It is usually easier to code in "high-level" languages than in "low-level" ones.