Also, specific user environment and usage history can make it difficult to reproduce the problem..  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
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.  
Many applications use a mix of several languages in their construction and use.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
Scripting and breakpointing is also part of this process.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
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.