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. However, readability is more than just programming style. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. One approach popular for requirements analysis is Use Case analysis. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. There are many approaches to the Software development process. 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. Programs were mostly entered using punched cards or paper tape. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Integrated development environments (IDEs) aim to integrate all such help. Many applications use a mix of several languages in their construction and use. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. It is very difficult to determine what are the most popular modern programming languages. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Normally the first step in debugging is to attempt to reproduce the problem. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. However, readability is more than just programming style. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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.