There are many approaches to the Software development process. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. 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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). One approach popular for requirements analysis is Use Case analysis. It is very difficult to determine what are the most popular modern programming languages. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Different programming languages support different styles of programming (called programming paradigms). A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. One approach popular for requirements analysis is Use Case analysis. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. It is very difficult to determine what are the most popular modern programming languages.