In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. There are many approaches to the Software development process. 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. 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. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. There exist a lot of different approaches for each of those tasks. Ideally, the programming language best suited for the task at hand will be selected. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. One approach popular for requirements analysis is Use Case analysis. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. Also, specific user environment and usage history can make it difficult to reproduce the problem. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.