In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Normally the first step in debugging is to attempt to reproduce the problem. There exist a lot of different approaches for each of those tasks. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Many applications use a mix of several languages in their construction and use. Scripting and breakpointing is also part of this process. Whatever the approach to development may be, the final program must satisfy some fundamental properties. It is very difficult to determine what are the most popular modern programming languages. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Following a consistent programming style often helps readability. Following a consistent programming style often helps readability. 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. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. 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.