For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. Scripting and breakpointing is also part of this process. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Programmable devices have existed for centuries. 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. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. 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. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Programs were mostly entered using punched cards or paper tape. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Following a consistent programming style often helps readability. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. One approach popular for requirements analysis is Use Case analysis. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Programs were mostly entered using punched cards or paper tape. There are many approaches to the Software development process. Different programming languages support different styles of programming (called programming paradigms). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.