Computer programmers are those who write computer software. Scripting and breakpointing is also part of this process. Code-breaking algorithms have also existed for centuries. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. However, readability is more than just programming style. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Programming languages are essential for software development. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. One approach popular for requirements analysis is Use Case analysis. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Code-breaking algorithms have also existed for centuries. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. Normally the first step in debugging is to attempt to reproduce the problem. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Scripting and breakpointing is also part of this process. There exist a lot of different approaches for each of those tasks. 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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.