Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. One approach popular for requirements analysis is Use Case analysis. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. It is usually easier to code in "high-level" languages than in "low-level" ones. 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). They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. However, readability is more than just programming style. However, readability is more than just programming style. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. 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. 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. There are many approaches to the Software development process. It affects the aspects of quality above, including portability, usability and most importantly maintainability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. However, readability is more than just programming style. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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.