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. 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. There are many approaches to the Software development process. Integrated development environments (IDEs) aim to integrate all such help. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Programming languages are essential for software development. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. Normally the first step in debugging is to attempt to reproduce the problem. Following a consistent programming style often helps readability. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Scripting and breakpointing is also part of this process. Many applications use a mix of several languages in their construction and use. Unreadable code often leads to bugs, inefficiencies, and duplicated code. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Scripting and breakpointing is also part of this process. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Also, specific user environment and usage history can make it difficult to reproduce the problem.