Programmable devices have existed for centuries. 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. Ideally, the programming language best suited for the task at hand will be selected. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It is very difficult to determine what are the most popular modern programming languages. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Scripting and breakpointing is also part of this process. Following a consistent programming style often helps readability. 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. 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. Scripting and breakpointing is also part of this process. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Following a consistent programming style often helps readability. Code-breaking algorithms have also existed for centuries. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. One approach popular for requirements analysis is Use Case analysis. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Programmable devices have existed for centuries. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.