Unreadable code often leads to bugs, inefficiencies, and duplicated code. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Normally the first step in debugging is to attempt to reproduce the problem. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. However, readability is more than just programming style. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. Code-breaking algorithms have also existed for centuries. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Techniques like Code refactoring can enhance readability. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Use of a static code analysis tool can help detect some possible problems. 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. There exist a lot of different approaches for each of those tasks. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Scripting and breakpointing is also part of this process. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.