The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. There are many approaches to the Software development process. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Many applications use a mix of several languages in their construction and use. Integrated development environments (IDEs) aim to integrate all such help. 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). 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. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. There are many approaches to the Software development process. 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. Techniques like Code refactoring can enhance readability. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. There exist a lot of different approaches for each of those tasks. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. It is usually easier to code in "high-level" languages than in "low-level" ones.