

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. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Programs were mostly entered using punched cards or paper tape. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). There exist a lot of different approaches for each of those tasks. 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). While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Integrated development environments (IDEs) aim to integrate all such help. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. Many applications use a mix of several languages in their construction and use. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Following a consistent programming style often helps readability. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. One approach popular for requirements analysis is Use Case analysis. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.