

High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. It is usually easier to code in "high-level" languages than in "low-level" ones. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. 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. There exist a lot of different approaches for each of those tasks. Following a consistent programming style often helps readability. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Integrated development environments (IDEs) aim to integrate all such help. 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. Programming languages are essential for software development. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Also, specific user environment and usage history can make it difficult to reproduce the problem. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Use of a static code analysis tool can help detect some possible problems. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.