

Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. Many applications use a mix of several languages in their construction and use. 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. Following a consistent programming style often helps readability. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Normally the first step in debugging is to attempt to reproduce the problem. Scripting and breakpointing is also part of this process. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Ideally, the programming language best suited for the task at hand will be selected. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. However, readability is more than just programming style. 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. Programs were mostly entered using punched cards or paper tape. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. One approach popular for requirements analysis is Use Case analysis.