

By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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. Ideally, the programming language best suited for the task at hand will be selected. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Also, specific user environment and usage history can make it difficult to reproduce the problem. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Programming languages are essential for software development. Computer programmers are those who write computer software. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. However, readability is more than just programming style. Computer programmers are those who write computer software. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. 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. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. 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. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code.