

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). 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Unreadable code often leads to bugs, inefficiencies, and duplicated code. There exist a lot of different approaches for each of those tasks. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. However, readability is more than just programming style. 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. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It is usually easier to code in "high-level" languages than in "low-level" ones. 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). After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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.