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. Many applications use a mix of several languages in their construction and use. 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. 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. 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. 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. 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. There are many approaches to the Software development process. Code-breaking algorithms have also existed for centuries. 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. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Following a consistent programming style often helps readability. Normally the first step in debugging is to attempt to reproduce the problem. However, readability is more than just programming style. Integrated development environments (IDEs) aim to integrate all such help. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. It is very difficult to determine what are the most popular modern programming languages. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Integrated development environments (IDEs) aim to integrate all such help. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.