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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Use of a static code analysis tool can help detect some possible problems. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Also, specific user environment and usage history can make it difficult to reproduce the problem. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Code-breaking algorithms have also existed for centuries. Integrated development environments (IDEs) aim to integrate all such help. Techniques like Code refactoring can enhance readability.