These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. 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. 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. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Techniques like Code refactoring can enhance readability. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Integrated development environments (IDEs) aim to integrate all such help. Programs were mostly entered using punched cards or paper tape. Integrated development environments (IDEs) aim to integrate all such help. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. However, readability is more than just programming style. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Ideally, the programming language best suited for the task at hand will be selected.