Use of a static code analysis tool can help detect some possible problems. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Computer programmers are those who write computer software. 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. 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. Programs were mostly entered using punched cards or paper tape. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Use of a static code analysis tool can help detect some possible problems. Also, specific user environment and usage history can make it difficult to reproduce the problem. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Ideally, the programming language best suited for the task at hand will be selected. Use of a static code analysis tool can help detect some possible problems. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Different programming languages support different styles of programming (called programming paradigms). However, Charles Babbage had already written his first program for the Analytical Engine in 1837.