

Following a consistent programming style often helps readability. Code-breaking algorithms have also existed for centuries. One approach popular for requirements analysis is Use Case analysis. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. There are many approaches to the Software development process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. 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. Techniques like Code refactoring can enhance readability. There exist a lot of different approaches for each of those tasks. Scripting and breakpointing is also part of this process. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Techniques like Code refactoring can enhance readability. Programs were mostly entered using punched cards or paper tape.