

The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. It affects the aspects of quality above, including portability, usability and most importantly maintainability. It is very difficult to determine what are the most popular modern programming 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. 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. Following a consistent programming style often helps readability. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. Code-breaking algorithms have also existed for centuries. Scripting and breakpointing is also part of this process. However, readability is more than just programming style. 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. 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. There are many approaches to the Software development process. 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.