

In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. 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. 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. However, readability is more than just programming style. Programming languages are essential for software development. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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. Use of a static code analysis tool can help detect some possible problems. Many applications use a mix of several languages in their construction and use. 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. 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. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.