

Unreadable code often leads to bugs, inefficiencies, and duplicated code. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. Scripting and breakpointing is also part of this process. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Normally the first step in debugging is to attempt to reproduce the problem. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Code-breaking algorithms have also existed for centuries. Integrated development environments (IDEs) aim to integrate all such help. One approach popular for requirements analysis is Use Case analysis. 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*. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.