

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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.

Code-breaking algorithms have also existed for centuries. One approach popular for requirements analysis is Use Case analysis. 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. Ideally, the programming language best suited for the task at hand will be selected. Programs were mostly entered using punched cards or paper tape. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. 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. 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.

Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.

Programmable devices have existed for centuries. 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. Scripting and breakpointing is also part of this process. Computer programmers are those who write computer software. Use of a static code analysis tool can help detect some possible problems. 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. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. Different programming languages support different styles of programming (called programming paradigms). It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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*.