It is usually easier to code in "high-level" languages than in "low-level" ones. 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. 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. 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. Programs were mostly entered using punched cards or paper tape. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Integrated development environments (IDEs) aim to integrate all such help. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Whatever the approach to development may be, the final program must satisfy some fundamental properties. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. It is very difficult to determine what are the most popular modern programming languages. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Ideally, the programming language best suited for the task at hand will be selected. Integrated development environments (IDEs) aim to integrate all such help. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Following a consistent programming style often helps readability. 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. 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.