Computer programmers are those who write computer software. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Programs were mostly entered using punched cards or paper tape. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. 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. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. 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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. 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. There exist a lot of different approaches for each of those tasks. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.