Programs were mostly entered using punched cards or paper tape. 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. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Ideally, the programming language best suited for the task at hand will be selected. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. However, readability is more than just programming style. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. 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). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Whatever the approach to development may be, the final program must satisfy some fundamental properties.