One approach popular for requirements analysis is Use Case analysis. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Programming languages are essential for software development. However, readability is more than just programming style. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. 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). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. Techniques like Code refactoring can enhance readability. One approach popular for requirements analysis is Use Case analysis. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).