Programming languages are essential for software development. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. Techniques like Code refactoring can enhance readability. 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. 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. 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. 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. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Also, specific user environment and usage history can make it difficult to reproduce the problem. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.