In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Programmable devices have existed for centuries. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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). Different programming languages support different styles of programming (called programming paradigms). Normally the first step in debugging is to attempt to reproduce the problem. Following a consistent programming style often helps readability. Ideally, the programming language best suited for the task at hand will be selected. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. 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). Computer programmers are those who write computer software. Unreadable code often leads to bugs, inefficiencies, and duplicated code. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.