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). Following a consistent programming style often helps readability. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Normally the first step in debugging is to attempt to reproduce the problem. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. However, readability is more than just programming style. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Ideally, the programming language best suited for the task at hand will be selected. There are many approaches to the Software development process. One approach popular for requirements analysis is Use Case analysis. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Programs were mostly entered using punched cards or paper tape. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Also, specific user environment and usage history can make it difficult to reproduce the problem. It is usually easier to code in "high-level" languages than in "low-level" ones. It is very difficult to determine what are the most popular modern programming languages. However, readability is more than just programming style. 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. 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.