There are many approaches to the Software development process. It affects the aspects of quality above, including portability, usability and most importantly maintainability. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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). Scripting and breakpointing is also part of this process. 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). They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Programmable devices have existed for centuries. Use of a static code analysis tool can help detect some possible problems. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. 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). Programming languages are essential for software development. 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. 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. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.