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. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. 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. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Ideally, the programming language best suited for the task at hand will be selected. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. Code-breaking algorithms have also existed for centuries. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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 compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Code-breaking algorithms have also existed for centuries.