However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Following a consistent programming style often helps readability. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Different programming languages support different styles of programming (called programming paradigms). Programming languages are essential for software development. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Different programming languages support different styles of programming (called programming paradigms). In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. One approach popular for requirements analysis is Use Case analysis. Programming languages are essential for software development. One approach popular for requirements analysis is Use Case analysis. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.