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. 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. 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. Different programming languages support different styles of programming (called programming paradigms). Scripting and breakpointing is also part of this process. 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. 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. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Use of a static code analysis tool can help detect some possible problems. Ideally, the programming language best suited for the task at hand will be selected. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Programs were mostly entered using punched cards or paper tape. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Scripting and breakpointing is also part of this process. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Programming languages are essential for software development. 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. There exist a lot of different approaches for each of those tasks. 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.