

However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Use of a static code analysis tool can help detect some possible problems. Code-breaking algorithms have also existed for centuries. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Also, specific user environment and usage history can make it difficult to reproduce the problem. One approach popular for requirements analysis is Use Case analysis. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Use of a static code analysis tool can help detect some possible problems. 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. 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. 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. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. Use of a static code analysis tool can help detect some possible problems. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Programming languages are essential for software development. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.