

Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. There are many approaches to the Software development process. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. Use of a static code analysis tool can help detect some possible problems. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). It affects the aspects of quality above, including portability, usability and most importantly maintainability. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. 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. 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. 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. It is very difficult to determine what are the most popular modern programming 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). Different programming languages support different styles of programming (called programming paradigms). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. One approach popular for requirements analysis is Use Case analysis. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.