

While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Programs were mostly entered using punched cards or paper tape. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Integrated development environments (IDEs) aim to integrate all such help. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Following a consistent programming style often helps readability. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.