

Ideally, the programming language best suited for the task at hand will be selected. 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. One approach popular for requirements analysis is Use Case analysis. 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. Code-breaking algorithms have also existed for centuries. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Programming languages are essential for software development. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Different programming languages support different styles of programming (called programming paradigms). 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. It is usually easier to code in "high-level" languages than in "low-level" ones. Unreadable code often leads to bugs, inefficiencies, and duplicated code. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. Scripting and breakpointing is also part of this process. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Programs were mostly entered using punched cards or paper tape. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).