

In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. It is very difficult to determine what are the most popular modern programming languages. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Use of a static code analysis tool can help detect some possible problems. Different programming languages support different styles of programming (called programming paradigms). The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Programs were mostly entered using punched cards or paper tape. Use of a static code analysis tool can help detect some possible problems. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Programmable devices have existed for centuries. Ideally, the programming language best suited for the task at hand will be selected. 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. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Code-breaking algorithms have also existed for centuries. It is usually easier to code in "high-level" languages than in "low-level" ones. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).