

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. Scripting and breakpointing is also part of this process. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming 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. 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. Ideally, the programming language best suited for the task at hand will be selected. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Ideally, the programming language best suited for the task at hand will be selected. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. There exist a lot of different approaches for each of those tasks. 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). After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Also, specific user environment and usage history can make it difficult to reproduce the problem. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Programs were mostly entered using punched cards or paper tape. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It is very difficult to determine what are the most popular modern programming languages. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Integrated development environments (IDEs) aim to integrate all such help.