

Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix 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. 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. Scripting and breakpointing is also part of this process. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Many applications use a mix of several languages in their construction and use. 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. Techniques like Code refactoring can enhance readability. It is usually easier to code in "high-level" languages than in "low-level" ones. One approach popular for requirements analysis is Use Case analysis. 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. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Techniques like Code refactoring can enhance readability. It is very difficult to determine what are the most popular modern programming languages. However, readability is more than just programming style. 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. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Different programming languages support different styles of programming (called programming paradigms). 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. 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. Many applications use a mix of several languages in their construction and use.