

Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Also, specific user environment and usage history can make it difficult to reproduce the problem. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. There exist a lot of different approaches for each of those tasks. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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. 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. 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).