

After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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.

Programmable devices have existed for centuries. It is very difficult to determine what are the most popular modern programming languages. Also, specific user environment and usage history can make it difficult to reproduce the problem. It is very difficult to determine what are the most popular modern programming languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Also, specific user environment and usage history can make it difficult to reproduce the problem. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Normally the first step in debugging is to attempt to reproduce the problem. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. 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. 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. Normally the first step in debugging is to attempt to reproduce the problem. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary 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. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. 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).