

This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. 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. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Unreadable code often leads to bugs, inefficiencies, and duplicated code. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. 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. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. There exist a lot of different approaches for each of those tasks. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. It is very difficult to determine what are the most popular modern programming languages. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.