

One approach popular for requirements analysis is Use Case analysis. 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). It is very difficult to determine what are the most popular modern programming languages. 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. Many applications use a mix of several languages in their construction and use. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Code-breaking algorithms have also existed for centuries. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. There are many approaches to the Software development process. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Normally the first step in debugging is to attempt to reproduce the problem. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. 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.