Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Also, specific user environment and usage history can make it difficult to reproduce the problem. Following a consistent programming style often helps readability. Programming languages are essential for software development. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. 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. 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. Ideally, the programming language best suited for the task at hand will be selected. However, readability is more than just programming style. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. One approach popular for requirements analysis is Use Case analysis. It is very difficult to determine what are the most popular modern programming languages. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. 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. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.