# DOWNLOAD PDF 4TH GENERATION PROGRAMMING LANGUAGE

## Chapter 1 : What is a Fourth Generation Programming Language (4GL)? - Definition from Techopedia

*A 4th-generation programming language (4GL) or (procedural language) is any computer programming language that belongs to a class of languages envisioned as an advancement upon third-generation programming languages (3GL).*

Humans communicate with one another in some language, like English, German or in many other languages. We, humans, can also communicate through gestures, facial expressions, even through our emotions we can express our selves and our feelings. In order to make computers work for us, some sort of instructions must be stored in a some kind of language. And that language is called a Programming Language. A programming language consists of all the symbols, characters, and usage rules that permit people to communicate with computers. There are at least several hundred, and possibly several thousand different programming languages. Some of these are created to serve a special purpose controlling a robot , while others are more flexible general-purpose tools that are suitable for many types of applications. What is a Programming Language? A programming language is a set of written symbols that instructs the computer hardware to perform specific tasks. Typically, a programming language consists of a vocabulary and a set of rules called syntax that the programmer must learn". It is a language made up of entirely 1s and 0s. There is not, however, one universal machine language because the language must be written in accordance with the special characteristics of a given processor. Each type or family of processor requires its own machine language. For this reason, machine language is said to be machine-dependent also called hardware-dependent. Machine language programs have the advantage of very fast execution speeds and efficient use of primary memory. Use of machine language is very tedious, difficult and time consuming method of programming. Machine language is low-level language. Since the programmer must specify every detail of an operation, a low-level language requires that the programmer have detailed knowledge of how the computer works. As a result, programmers were few in numbers and lacked complexity. To make programming simpler, other easier-to-use programming languages have been developed. These languages, however must ultimately be translated into machine language before the computer can understand and use them. They are also classified as low-level languages because detailed knowledge of hardware is still required. They were developed in s. Assembly languages use mnemonic operation codes and symbolic addresses in place of 1s and 0s to represent the operation codes. A mnemonic is an alphabetical abbreviation used as memory aid. This means a programmer can use abbreviation instead of having to remember lengthy binary instruction codes. For example, it is much easier to remember L for Load, A for Add, B for Branch, and C for Compare than the binary equivalents i-e different combinations of 0s and 1s. Assembly language uses symbolic addressing capabilities that simplify the programming process because the programmer does not need to know or remember the exact storage locations of instructions or data. Symbolic addressing is the ability to express an address in terms of symbols chosen by the programmer rather than in terms of the absolute numerical location. Therefore, it is not necessary to assign and remember a number that identifies the address of a piece of data. Although assembly languages represented an improvement, they had obvious limitations. Only computer specialists familiar with the architecture of the computer being used can use them. And because they are also machine dependent, assembly languages are not easily converted to run on other types of computers. Before they can be used by the computer, assembly languages must be translated into machine language. A language translator program called an assembler does this conversion. Assembly languages provide an easier and more efficient way to program than machine languages while still maintaining control over the internal functions of a computer at the most basic level. The advantages of programming with assembly languages are that they produce programs that are efficient, use less storage, and execute much faster than programs designed using high-level languages. They are designed to run on a number of different computers with few or no changes. Objectives of high-level languages To relieve the programmer of the detailed and tedious task of writing programs in machine language and assembly languages. To provide programs that can be used on more than one type of machine with very few changes. Most high level languages are considered to be procedure-oriented, or Procedural languages, because the program instructions comprise lists of steps, procedures, that tell the computer not only what to do but how

to do it. High-level language statements generate, when translated, a comparatively greater number of assembly language instructions and even more machine language instructions. The programmer spends less time developing software with a high level language than with assembly or machine language because fewer instructions have to be created. A language translator is required to convert a high-level language program into machine language. Two types of language translators are used with high level languages: They are non-procedural languages, so named because they allow programmers and users to specify what the computer is supposed to do without having to specify how the computer is supposed to do it. Consequently, fourth generation languages need approximately one tenth the number of statements that a high level languages needs to achieve the same results. Because they are so much easier to use than third generation languages, fourth generation languages allow users, or non-computer professionals, to develop software. Objectives of fourth generation languages Increasing the speed of developing programs. Minimizing user effort to obtain information from computer. Decreasing the skill level required of users so that they can concentrate on the application rather than the intricacies of coding, and thus solve their own problems without the aid of a professional programmer. Minimizing maintenance by reducing errors and making programs that are easy to change. Depending on the language, the sophistication of fourth generation languages varies widely. These languages are usually used in conjunction with a database and its data dictionary. Five basic types of language tools fall into the fourth generation language category.

## Chapter 2 : Forth (programming language) - Wikipedia

*A fourth generation (programming) language (4GL) is a grouping of programming languages that attempt to get closer than 3GLs to human language, form of thinking and conceptualization. 4GLs are designed to reduce the overall time, effort and cost of software development.*

Computer â€" A computer is a device that can be instructed to carry out an arbitrary set of arithmetic or logical operations automatically. The ability of computers to follow a sequence of operations, called a program, such computers are used as control systems for a very wide variety of industrial and consumer devices. The Internet is run on computers and it millions of other computers. Since ancient times, simple manual devices like the abacus aided people in doing calculations, early in the Industrial Revolution, some mechanical devices were built to automate long tedious tasks, such as guiding patterns for looms. More sophisticated electrical machines did specialized analog calculations in the early 20th century, the first digital electronic calculating machines were developed during World War II. The speed, power, and versatility of computers has increased continuously and dramatically since then, conventionally, a modern computer consists of at least one processing element, typically a central processing unit, and some form of memory. Peripheral devices allow information to be retrieved from an external source and this usage of the term referred to a person who carried out calculations or computations. The word continued with the same meaning until the middle of the 20th century, from the end of the 19th century the word began to take on its more familiar meaning, a machine that carries out computations. The Online Etymology Dictionary gives the first attested use of computer in the s, one who calculates, the Online Etymology Dictionary states that the use of the term to mean calculating machine is from  The Online Etymology Dictionary indicates that the use of the term. The earliest counting device was probably a form of tally stick, later record keeping aids throughout the Fertile Crescent included calculi which represented counts of items, probably livestock or grains, sealed in hollow unbaked clay containers. The use of counting rods is one example, the abacus was initially used for arithmetic tasks. The Roman abacus was developed from used in Babylonia as early as BC. Since then, many forms of reckoning boards or tables have been invented. In a medieval European counting house, a checkered cloth would be placed on a table, the Antikythera mechanism is believed to be the earliest mechanical analog computer, according to Derek J. It was designed to calculate astronomical positions and it was discovered in in the Antikythera wreck off the Greek island of Antikythera, between Kythera and Crete, and has been dated to circa BC 2. Sperry Corporation â€" Sperry Corporation was a major American equipment and electronics company whose existence spanned more than seven decades of the 20th century. Through a series of mergers it exists today as a part of Unisys, during World War I the company diversified into aircraft components including bomb sights and fire control systems. In their early decades, Sperry Gyroscope and related companies were concentrated on Long Island, New York, over the years, it diversified to other locations. In , Lawrence Sperry split from his father to compete over aero-instruments with the Lawrence Sperry Aircraft Company, in , following the death of Lawrence on December 13,, the two firms were brought together. The company became Sperry Corporation in , the new corporation was a holding company for a number of smaller entities such as the original Sperry Gyroscope, Ford Instrument Company, Intercontinental Aviation, Inc. The company made advanced aircraft navigation equipment for the market, including the Sperry Gyroscope, the company prospered during World War II as military demand skyrocketed, ranking 19th among US corporations in the value of wartime production contracts. It specialized in high technology such as analog computerâ€"controlled bomb sights, airborne radar systems. This was to preserve parts of this company in the event of a thermonuclear conflagration. The company remained a military contractor. From to the corporation was involved in an acrimonious antitrust lawsuit with Honeywell, the company dropped Rand from its title and reverted to Sperry Corporation. At about the time as the Rand acquisition, Sperry Gyroscope decided to open a facility that would almost exclusively produce its marine instruments. After considerable searching and evaluation, a plant was built in Charlottesville, Virginia and it was later renamed Sperry Marine. This group is now part of Lockheed Martin, Sperry in Britain started with a factory in Pimlico, London in , manufacturing

gyroscopic compasses for the Royal Navy. Punched card â€" A punched card or punch card is a piece of stiff paper that can be used to contain digital information represented by the presence or absence of holes in predefined positions. The information might be data for data processing applications or, in earlier examples, the terms IBM card, or Hollerith card specifically refer to punched cards used in semiautomatic data processing. Many early digital computers used punched cards, often prepared using keypunch machines, while punched cards are now obsolete as a recording medium, as of , some voting machines still use punched cards to record votes. Basile Bouchon developed the control of a loom by punched holes in paper tape in , in Joseph Marie Jacquard demonstrated a mechanism to automate loom operation. A number of punched cards were linked into a chain of any length, each card held the instructions for shedding and selecting the shuttle for a single pass. It is considered an important step in the history of computing hardware, semen Korsakov was reputedly the first to use the punched cards in informatics for information store and search. Korsakov announced his new method and machines in September , rather than seeking patents, charles Babbage proposed the use of Number Cards, pierced with certain holes and stand opposite levers connected with a set of figure wheels. Advanced they push in those levers opposite to which there are no holes on the card, Herman Hollerith invented the recording of data on a medium that could then be read by a machine. Prior uses of machine readable media, such as those above, had been for control, after some initial trials with paper tape, he settled on punched cards. Developing punched card data processing technology for the US census, other companies entering the punched card business included the Powers Accounting Machine Company, Remington Rand, and Groupe Bull. Both IBM and Remington Rand tied punched card purchases to machine leases, in , the US government took both to court on this issue. IBM viewed its business as providing a service and that the cards were part of the machine, IBM fought all the way to the Supreme Court and lost in , the court ruling that IBM could only set card specifications. Punched cards were used as legal documents, such as U. Punched card technology developed into a tool for business data-processing. By punched cards had become ubiquitous in industry and government, do not fold, spindle or mutilate, a generalized version of the warning that appeared on some punched cards, became a motto for the post-World War II era. The UNITYPER introduced magnetic tape for data entry in the s, during the s, the punched card was gradually replaced as the primary means for data storage by magnetic tape, as better, more capable computers became available 4. Unisys â€" Unisys Corporation is an American global information technology company based in Blue Bell, Pennsylvania, that provides a portfolio of IT services, software, and technology. In September Unisys was formed through the merger of the mainframe corporations Sperry and Burroughs, the name was chosen from over 31, submissions in an internal competition when Chuck Ayoub submitted the word Unisys which was composed of parts of the words united, information and systems. The merger was the largest in the industry at the time. At the time of the merger, Unisys had approximately , employees, michael Blumenthal became CEO and Chairman after the merger and resigned in after several years of losses. James Unruh, became the new CEO and Chairman after Blumenthals departure and continued in that role until , by , layoffs had reduced world-wide employee count to approximately 30, In addition to hardware, both Burroughs and Sperry had a history of working on U. On October 7,, J. Edward Coleman replaced J. On October 6,, Unisys announced that Coleman would leave the company effective December 1,, Unisys share price immediately fell when this news became public. In , Unisys phased out its CMOS processors, completing the migration of its ClearPath mainframes to Intel x86 chips, the company announced its new ClearPath Dorado and systems in May,, its most powerful Dorado systems ever. In , CRN ranked Unisys Stealth on its list of Top 10 products for combatting advanced persistent threat, Platform among the Top 10 coolest servers of 5. To create a demand for its services, the set up real estate offices. Despite the name, its main line never served Santa Fe, New Mexico, as the terrain was too difficult, the Santa Fe was a pioneer in intermodal freight transport, an enterprise that included a tugboat fleet and an airline. Building across Kansas and eastern Colorado was simple, with few natural obstacles and it set up real estate offices in the area and promoted settlement across Kansas on the land that was granted to it by Congress in  It offered discounted fares to anyone who traveled west to land, if the land was purchased. The line that was to become the St. Louis, Missouri for the Frisco, but the Tulsa-Albuquerque portion remained unbuilt, by January , the entire system consisted of some 7, miles of

track 6. Computer hardware â€" Computer hardware is the collection of physical components that constitute a computer system. By contrast, software is instructions that can be stored and run by hardware, hardware is directed by the software to execute any command or instruction. A combination of hardware and software forms a usable computing system, the template for all modern computers is the Von Neumann architecture, detailed in a paper by Hungarian mathematician John von Neumann. The meaning of the term has evolved to mean a stored-program computer in which an instruction fetch and this is referred to as the Von Neumann bottleneck and often limits the performance of the system. For the third year, U. The impressive growth was the fastest sales increase since the end of the recession, sales growth accelerated in the second half of the year peaking in fourth quarter with a 6. There are a number of different types of system in use today. The personal computer, also known as the PC, is one of the most common types of computer due to its versatility, laptops are generally very similar, although they may use lower-power or reduced size components, thus lower performance. The computer case is a plastic or metal enclosure that houses most of the components, a case can be either big or small, but the form factor of motherboard for which it is designed matters more. A power supply unit converts alternating current electric power to low-voltage DC power for the components of the computer. Laptops are capable of running from a battery, normally for a period of hours. The motherboard is the component of a computer. It is usually cooled by a heatsink and fan, or water-cooling system, most newer CPUs include an on-die Graphics Processing Unit. The clock speed of CPUs governs how fast it executes instructions, many modern computers have the option to overclock the CPU which enhances performance at the expense of greater thermal output and thus a need for improved cooling. The chipset, which includes the bridge, mediates communication between the CPU and the other components of the system, including main memory. Statistics â€" Statistics is a branch of mathematics dealing with the collection, analysis, interpretation, presentation, and organization of data. In applying statistics to, e. Statistics deals with all aspects of data including the planning of data collection in terms of the design of surveys, statistician Sir Arthur Lyon Bowley defines statistics as Numerical statements of facts in any department of inquiry placed in relation to each other. When census data cannot be collected, statisticians collect data by developing specific experiment designs, representative sampling assures that inferences and conclusions can safely extend from the sample to the population as a whole. In contrast, an observational study does not involve experimental manipulation, inferences on mathematical statistics are made under the framework of probability theory, which deals with the analysis of random phenomena. A standard statistical procedure involves the test of the relationship between two data sets, or a data set and a synthetic data drawn from idealized model. A hypothesis is proposed for the relationship between the two data sets, and this is compared as an alternative to an idealized null hypothesis of no relationship between two data sets. Rejecting or disproving the hypothesis is done using statistical tests that quantify the sense in which the null can be proven false. Working from a hypothesis, two basic forms of error are recognized, Type I errors and Type II errors. Multiple problems have come to be associated with this framework, ranging from obtaining a sufficient sample size to specifying an adequate null hypothesis, measurement processes that generate statistical data are also subject to error. Many of these errors are classified as random or systematic, the presence of missing data or censoring may result in biased estimates and specific techniques have been developed to address these problems. Statistics continues to be an area of research, for example on the problem of how to analyze Big data. Statistics is a body of science that pertains to the collection, analysis, interpretation or explanation. Some consider statistics to be a mathematical science rather than a branch of mathematics. While many scientific investigations make use of data, statistics is concerned with the use of data in the context of uncertainty, mathematical techniques used for this include mathematical analysis, linear algebra, stochastic analysis, differential equations, and measure-theoretic probability theory. In applying statistics to a problem, it is practice to start with a population or process to be studied. Populations can be diverse topics such as all living in a country or every atom composing a crystal. Ideally, statisticians compile data about the entire population and this may be organized by governmental statistical institutes 8. Computer programming in the punched card era â€" From the invention of computer programming languages up to the mids, many if not most computer programmers created, edited and stored their programs line by line on punched cards. The practice was nearly universal with

IBM computers in the era, a punched card is a flexible write-once medium that encodes data, most commonly 80 characters. Groups or decks of cards form programs and collections of data, users could create cards using a desk-sized keypunch with a typewriter-like keyboard. A typing error generally necessitated repunching an entire card and these forms were then converted to cards by keypunch operators, and in some cases, checked by verifiers.

## Chapter 3 : fourth-generation language Definition from PC Magazine Encyclopedia

*No single 4GL in the mainstream of the s lived up to the hype. The writing was on the wall. Languages well classified as 3GL were going to be the direction for the foreseeable future. Classic 4GL Visual Basic set sail thousands of ideas in corporate departments everywhere. VB infused freeware.*

Fourth-generation programming language Save A 4th-generation programming language 4GL or procedural language is any computer programming language that belongs to a class of languages envisioned as an advancement upon third-generation programming languages 3GL. Each of the programming language generations aims to provide a higher level of abstraction of the internal computer hardware details, making the language more programmer -friendly, powerful and versatile. While the definition of 4GL has changed over time, it can be typified by operating more with large collections of information at once rather than focusing on just bits and bytes. Languages claimed to be 4GL may include support for database management, report generation , mathematical optimization , GUI development , or web development. Some researchers state that 4GLs are a subset of domain-specific languages. Also, libraries with 4GL-like features have been developed as add-ons for most popular 3GLs. This has blurred the distinction of 4GL and 3GL. In the s and s, there were efforts to develop fifth-generation programming languages 5GL. History Though used earlier in papers and discussions, the term 4GL was first used formally by James Martin in his book Applications Development Without Programmers[3] to refer to non-procedural, high-level specification languages. The term can apply to a large set of software products. It can also apply to an approach that looks for greater semantic properties and implementation power. Just as the 3GL offered greater power to the programmer, so too did the 4GL open up the development environment to a wider population. With judicious use of a few cards, the 4GL deck could offer a wide variety of processing and reporting capability whereas the equivalent functionality coded in a 3GL could subsume, perhaps, a whole box or more of cards. Even with its limitations, this approach supported highly sophisticated applications. As interfaces improved and allowed longer statement lengths and grammar-driven input handling, greater power ensued. An example of this is described on the Nomad page. The development of the 4GL was influenced by several factors, with the hardware and operating system constraints having a large weight. When the 4GL was first introduced, a disparate mix of hardware and operating systems mandated custom application development support that was specific to the system in order to ensure sales. Though it has roots back to the beginning, the system has proven successful in many applications and has been ported to modern platforms. The latest variant is embedded in the BIS[5] offering of Unisys. Cohen at Mathematica, a mathematical software company. Later 4GL types are tied to a database system and are far different from the earlier types in their use of techniques and resources that have resulted from the general improvement of computing with time. Types A number of different types of 4GLs exist: Table-driven codeless programming, usually running with a runtime framework and libraries. Instead of using code, the developer defines their logic by selecting an operation in a pre-defined list of memory or data table manipulation commands. In other words, instead of coding, the developer uses table-driven algorithm programming see also control tables that can be used for this purpose. A good example of this type of 4GL language is PowerBuilder. These types of tools can be used for business application development usually consisting in a package allowing for both business data manipulation and reporting, therefore they come with GUI screens and report editors. Report-generator programming languages take a description of the data format and the report to generate and from that they either generate the required report directly or they generate a program to generate the report. See also RPG Similarly, forms generators manage online interactions with the application system users or generate programs to do so. More ambitious 4GLs sometimes termed fourth generation environments attempt to automatically generate whole systems from the outputs of CASE tools, specifications of screens and reports, and possibly also the specification of some additional processing logic. Data management 4GLs such as SAS , SPSS and Stata provide sophisticated coding commands for data manipulation, file reshaping, case selection and data documentation in the preparation of data for statistical analysis and reporting. Some 4GLs have integrated tools which allow for the easy specification of all the

required information: Low code environments In the twenty-first century, 4GL systems have emerged as "low code" environments or platforms for the problem of rapid application development in short periods of time. Vendors often provide sample systems such as CRM, contract management, bug tracking from which development can occur with little programming.

## Chapter 4 : Fourth-generation programming language | Revolvy

*Often abbreviated 4GL, fourth-generation languages are programming languages closer to human languages than typical high-level programming languages. Most 4GLs are used to access databases. For example, a typical 4GL commandis FIND ALL RECORDS WHERE NAME IS "SMITH" Stay up to date on the latest.*

Vacuum Tubes The first computer systems used vacuum tubes for circuitry and magnetic drums for memory , and were often enormous, taking up entire rooms. These computers were very expensive to operate and in addition to using a great deal of electricity, the first computers generated a lot of heat, which was often the cause of malfunctions. First generation computers relied on machine language , the lowest-level programming language understood by computers, to perform operations, and they could only solve one problem at a time. It would take operators days or even weeks to set-up a new problem. Input was based on punched cards and paper tape, and output was displayed on printouts. Census Bureau in  Transistors The world would see transistors replace vacuum tubes in the second generation of computers. The transistor was invented at Bell Labs in but did not see widespread use in computers until the late s. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation predecessors. Though the transistor still generated a great deal of heat that subjected the computer to damage, it was a vast improvement over the vacuum tube. Second-generation computers still relied on punched cards for input and printouts for output. From Binary to Assembly Second-generation computers moved from cryptic binary machine language to symbolic, or assembly , languages, which allowed programmers to specify instructions in words. These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology. The first computers of this generation were developed for the atomic energy industry. Integrated Circuits The development of the integrated circuit was the hallmark of the third generation of computers. Transistors were miniaturized and placed on silicon chips , called semiconductors , which drastically increased the speed and efficiency of computers. Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system , which allowed the device to run many different applications at one time with a central program that monitored the memory. Computers for the first time became accessible to a mass audience because they were smaller and cheaper than their predecessors. An integrated circuit IC is a small electronic device made out of a semiconductor material. Microprocessors Present The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip. What in the first generation filled an entire room could now fit in the palm of the hand. In IBM introduced its first computer for the home user, and in Apple introduced the Macintosh. Microprocessors also moved out of the realm of desktop computers and into many areas of life as more and more everyday products began to use microprocessors. As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet. Fourth generation computers also saw the development of GUIs , the mouse and handheld devices. Artificial Intelligence Present and Beyond Fifth generation computing devices, based on artificial intelligence , are still in development, though there are some applications, such as voice recognition , that are being used today. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation and molecular and nanotechnology will radically change the face of computers in years to come. The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

## Chapter 5 : The Five Generations of Computers - Webopedia Reference

*Second-generation languages are machine-dependent assembly languages, and third-generation languages (3GLs) are high-level programming languages, such as FORTRAN, COBOL, BASIC, Pascal, C/C++ and Java.*

History[ edit ] Though used earlier in papers and discussions, the term 4GL was first used formally by James Martin in his book Applications Development Without Programmers [3] to refer to non-procedural, high-level specification languages. The term can apply to a large set of software products. It can also apply to an approach that looks for greater semantic properties and implementation power. Just as the 3GL offered greater power to the programmer, so too did the 4GL open up the development environment to a wider population. With judicious use of a few cards, the 4GL deck could offer a wide variety of processing and reporting capability whereas the equivalent functionality coded in a 3GL could subsume, perhaps, a whole box or more of cards. Even with its limitations, this approach supported highly sophisticated applications. As interfaces improved and allowed longer statement lengths and grammar-driven input handling, greater power ensued. An example of this is described on the Nomad page. The development of the 4GL was influenced by several factors, with the hardware and operating system constraints having a large weight. When the 4GL was first introduced, a disparate mix of hardware and operating systems mandated custom application development support that was specific to the system in order to ensure sales. Though it has roots back to the beginning, the system has proven successful in many applications and has been ported to modern platforms. The latest variant is embedded in the BIS [5] offering of Unisys. Cohen at Mathematica, a mathematical software company. Later 4GL types are tied to a database system and are far different from the earlier types in their use of techniques and resources that have resulted from the general improvement of computing with time. Types[ edit ] A number of different types of 4GLs exist: Table-driven codeless programming, usually running with a runtime framework and libraries. Instead of using code, the developer defines their logic by selecting an operation in a pre-defined list of memory or data table manipulation commands. In other words, instead of coding, the developer uses table-driven algorithm programming see also control tables that can be used for this purpose. A good example of this type of 4GL language is PowerBuilder. These types of tools can be used for business application development usually consisting in a package allowing for both business data manipulation and reporting, therefore they come with GUI screens and report editors. Report-generator programming languages take a description of the data format and the report to generate and from that they either generate the required report directly or they generate a program to generate the report. See also RPG Similarly, forms generators manage online interactions with the application system users or generate programs to do so. More ambitious 4GLs sometimes termed fourth generation environments attempt to automatically generate whole systems from the outputs of CASE tools, specifications of screens and reports, and possibly also the specification of some additional processing logic. Data management 4GLs such as SAS , SPSS and Stata provide sophisticated coding commands for data manipulation, file reshaping, case selection and data documentation in the preparation of data for statistical analysis and reporting. Some 4GLs have integrated tools which allow for the easy specification of all the required information: Low code environments[ edit ] In the twenty-first century, 4GL systems have emerged as "low code" environments or platforms for the problem of rapid application development in short periods of time. Vendors often provide sample systems such as CRM, contract management, bug tracking from which development can occur with little programming.

## Chapter 6 : Fourth-generation programming language - Wikipedia

*A fourth-generation programming language (4GL) is a computer programming language envisioned as a refinement of the style of languages classified as third-generation programming language (3GL).*

Overview[ edit ] A Forth environment combines the compiler with an interactive shell, where the user defines and runs subroutines called words. Words can be tested, redefined, and debugged as the source is entered without recompiling or restarting the whole program. All syntactic elements, including variables and basic operators, are defined as words. Forth environments vary in how the resulting program is stored, but ideally running the program has the same effect as manually re-entering the source. Stacks[ edit ] Most programming environments with recursive subroutines use a stack for control flow. This structure typically also stores local variables , including subroutine parameters in call by value system such as C. Forth often does not have local variables, however, nor is it call-by-value. Instead, intermediate values are kept in another stack , different from the one it uses for return addresses, loop counters, etc. Words operate directly on the topmost values in the first of these two stacks. It may therefore be called the "parameter" or "data" stack, but most often simply "the" stack. The second, function-call stack is then called the "linkage" or "return" stack, abbreviated rstack. Special rstack manipulation functions provided by the kernel allow it to be used for temporary storage within a word, and it is often used by counted loops, but otherwise it cannot be used to pass parameters or manipulate data. Most words are specified in terms of their effect on the stack. Typically, parameters are placed on the top of the stack before the word executes. After execution, the parameters have been erased and replaced with any return values. For arithmetic operators, this follows the rule of reverse Polish notation. See below for examples illustrating stack usage. Maintenance[ edit ] Forth is a simple yet extensible language; its modularity and extensibility permit the writing of high-level programs such as CAD systems. Forth has been used successfully in large, complex projects, while applications developed by competent, disciplined professionals have proven to be easily maintained on evolving hardware platforms over decades of use. It has been implemented efficiently on modern RISC processors , and processors that use Forth as machine language have been produced. Forth is so named because in "the file holding the interpreter was labeled FOURTH, for 4th next generation softwareâ€”but the IBM operating system restricted file names to 5 characters. Because Charles Moore frequently moved from job to job over his career, an early pressure on the developing language was ease of porting to different computer architectures. A Forth system has often been used to bring up new hardware. Wide dissemination finally led to standardization of the language. A complete family tree is at TU-Wien. Reverse Polish notation Forth relies heavily on explicit use of a data stack and reverse Polish notation RPN or postfix notation , commonly used in calculators from Hewlett-Packard. In RPN, the operator is placed after its operands, as opposed to the more common infix notation where the operator is placed between its operands. Extending the compiler only requires writing a new word, instead of modifying a grammar and changing the underlying implementation. Then the number 50 is placed on the stack. The CR moves the output to a new line it is only for formatting purposes and could be omitted butâ€”in most implementationsâ€”without it the output would occur on the same line as the input and would be less readable in the example. As everything has completed successfully at that point, the text interpreter then outputs the prompt "ok" and moves to a new line to get more input without needing anything explicit to do that. The text in parentheses is a comment, advising that this word expects a number on the stack and will return a possibly changed number. The CR moves the output to a new line again, this is only here for readability. Finally, a call to ". Facilities[ edit ] Forth has no explicit grammar. The interpreter reads a line of input from the user input device, which is then parsed for a word using spaces as a delimiter ; some systems recognise additional whitespace characters. When the interpreter finds a word, it looks the word up in the dictionary. If the word is found, the interpreter executes the code associated with the word, and then returns to parse the rest of the input stream. Otherwise, if both the lookup and the number conversion fail, the interpreter prints the word followed by an error message indicating the word is not recognised, flushes the input stream, and waits for new user input. When executed by typing 10 X at the console this will print 11  Mostly the assembler is tucked away in

a separate namespace wordlist as relatively few users want to use it. Forth assemblers may use a reverse-polish syntax in which the parameters of an instruction precede the instruction, but designs vary widely and are specific to the Forth implementation. A typical reverse-polish assembler prepares the operands on the stack and have the mnemonic copy the whole instruction into memory as the last step. A Forth assembler is by nature a macro assembler, so that it is easy to define an alias for registers according to their role in the Forth system: All modern Forth systems use normal text files for source, even if they are embedded. An embedded system with a resident compiler gets its source via a serial line. Classic Forth systems traditionally use neither operating system nor file system. Instead of storing code in files, source code is stored in disk blocks written to physical disk addresses. The word BLOCK is employed to translate the number of a 1K-sized block of disk space into the address of a buffer containing the data, which is managed automatically by the Forth system. Block use has become rare since the mids. In a hosted system those blocks too are allocated in a normal file in any case. Multitasking , most commonly cooperative round-robin scheduling , is normally available although multitasking words and support are not covered by the ANSI Forth Standard. Each task has its own stacks, private copies of some control variables and a scratch area. The usual method is to redefine the handful of words that place compiled bits into memory. The buffer area simulates or accesses a memory area beginning at a different address than the code buffer. This effectively reuses all the code of the compiler and interpreter. The buffer in memory is written to disk, and ways are provided to load it temporarily into memory for testing. When the new version appears to work, it is written over the previous version. Numerous variations of such compilers exist for different environments. For embedded systems , the code may instead be written to another computer, a technique known as cross compilation , over a serial port or even a single TTL bit, while keeping the word names and other non-executing parts of the dictionary in the original compiling computer. The minimum definitions for such a Forth compiler are the words that fetch and store a byte, and the word that commands a Forth word to be executed. Often the most time-consuming part of writing a remote port is constructing the initial program to implement fetch, store and execute, but many modern microprocessors have integrated debugging features such as the Motorola CPU32 that eliminate this task. The dictionary is laid out in memory as a tree of linked lists with the links proceeding from the latest most recently defined word to the oldest, until a sentinel value , usually a NULL pointer, is found. A context switch causes a list search to start at a different leaf. A linked list search continues as the branch merges into the main trunk leading eventually back to the sentinel, the root. There can be several dictionaries. In rare cases such as meta-compilation a dictionary might be isolated and stand-alone. The effect resembles that of nesting namespaces and can overload keywords depending on the context. A defined word generally consists of head and body with the head consisting of the name field NF and the link field LF and body consisting of the code field CF and the parameter field PF. Head and body of a dictionary entry are treated separately because they may not be contiguous. For example, when a Forth program is recompiled for a new platform, the head may remain on the compiling computer, while the body goes to the new platform. In some environments such as embedded systems the heads occupy memory unnecessarily. However, some cross-compilers may put heads in the target if the target itself is expected to support an interactive Forth. However, certain components are almost always present, though the exact size and order may vary. Described as a structure, a dictionary entry might look this way: The link field contains a pointer to the previously defined word. The pointer may be a relative displacement or an absolute address that points to the next oldest sibling. The code field pointer will be either the address of the word which will execute the code or data in the parameter field or the beginning of machine code that the processor will execute directly. For colon defined words, the code field pointer points to the word that will save the current Forth instruction pointer IP on the return stack, and load the IP with the new address from which to continue execution of words. Structure of the compiler[ edit ] The compiler itself is not a monolithic program. It consists of Forth words visible to the system, and usable by a programmer. The "compile time" flag in the name field is set for words with "compile time" behavior. Most simple words execute the same code whether they are typed on a command line, or embedded in code. When compiling these, the compiler simply places code or a threaded pointer to the word. Counted loop control flow words work similarly but set up combinations of primitive words that work with a counter, and so on. During

compilation, the data stack is used to support control structure balancing, nesting, and back-patching of branch addresses. LIT is the primitive word for pushing a "literal" number onto the data stack. Compilation state and interpretation state[ edit ] The word: The interpreter continues to read space-delimited words from the user input device. If a word is found, the interpreter executes the compilation semantics associated with the word, instead of the interpretation semantics. The default compilation semantics of a word are to append its interpretation semantics to the current definition. It is an example of a word whose compilation semantics differ from the default. The interpretation semantics of ; semi-colon , most control flow words, and several other words are undefined in ANS Forth, meaning that they must only be used inside of definitions and not on the interactive command line. These words can be used with the word LITERAL to calculate a value during a compilation and to insert the calculated value into the current colon definition. LITERAL has the compilation semantics to take an object from the data stack and to append semantics to the current colon definition to place that object on the data stack. This allows the implementation of so-called state-smart words with behavior that changes according to the current state of the interpreter. Immediate words[ edit ] The word IMMEDIATE marks the most recent colon definition as an immediate word, effectively replacing its compilation semantics with its interpretation semantics. NONAME which compiles the following words up to the next ; semi-colon and leaves an execution token on the data stack. The execution token provides an opaque handle for the compiled semantics, similar to the function pointers of the C programming language.

## Chapter 7 : Fourth-generation programming language - WikiVisually

*Fourth-generation language (4GL), Fourth-generation computer programming language. 4GLs are closer to human language than other high-level languages and are accessible to people without formal training as programmers. They allow multiple common operations to be performed with a single programmer-entered command.*

## Chapter 8 : Eng | Project Gutenberg Self-Publishing - eBooks | Read eBooks online

*Moore saw Forth as a successor to compile-link-go third-generation programming languages, or software for "fourth generation" hardware, not a fourth-generation programming language as the term has come to be used.*

## Chapter 9 : Generation of programming languages

*Originally Answered: What are some examples of the 4th generation programming languages? You could think of program generators with their input commands and parameters as crude 4GL programming languages.*