DOWNLOAD PDF THE COMPUTER PRIMER

Chapter 1 : potpourri | primer

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

Additive mixing explains how light from these colored elements can be used for photorealistic color image reproduction. The perception elicited by multiple light sources co-stimulating the same area of the retina is additive, i. For example, a purple spotlight on a dark background could be matched with coincident blue and red spotlights that are both dimmer than the purple spotlight. If the intensity of the purple spotlight was doubled it could be matched by doubling the intensities of both the red and blue spotlights that matched the original purple. The original monochromatic primaries of the arbitrary wavelengths of It is important to note that additive mixing provides very poor predictions of color perception outside the color matching context. Well known demonstrations such as The dress and other examples [13] show how the additive mixing model alone is not sufficient for predicting perceived color in many instances of real images. In general, we cannot completely predict all possible perceived colors from combinations of primary lights in the context of real world images and viewing conditions. The cited examples suggest just how remarkably poor such predictions can be. Subtractive mixing of ink layers[edit] See also: CMYK color model A magnified representation of small partially overlapping spots of cyan, magenta, yellow, and key black halftones in CMYK process printing. Each row represents the pattern of partially overlapping ink "rosettes" so that the patterns would be perceived as blue, green, and red when viewed on white paper from a typical viewing distance. The overlapping ink layers mix subtractively while additive mixing predicts the color appearance from the light reflected from the rosettes and white paper in between them. The subtractive color mixing model predicts the spectral power distributions of light filtered through overlaid partially absorbing materials on a reflecting or transparent surface. Each layer partially absorbs some wavelengths of light from the illumination spectrum while letting others pass through multiplicatively, resulting in a colored appearance. Overlapping layers of ink in printing mix subtractively over reflecting white paper in this way to generate photorealistic color images. The typical number of inks in such a printing process ranges from 3 to 6 e. In general, using fewer inks as primaries results in more economical printing but using more may result in better color reproduction. Cyan , magenta , and yellow are good subtractive primaries in that the spectral power distributions of light reflected from idealized inks can be combined for the largest chromaticity gamuts. Before the color names cyan and magenta were in common use, these primaries were often known as blue and red, respectively, and their exact color has changed over time with access to new pigments and technologies. Color predictions that incorporate light scattering effects of pigment particles and paint layer thickness require approaches based on the Kubelkaâ€”Munk [18] equations. Even such approaches cannot predict the color of paint mixtures precisely since small variances in particle size distribution, impurity concentrations etc. Artists typically rely on mixing experience and "recipes" [19] to mix desired colors from a small initial set of primaries and do not use mathematical modelling. There are hundreds of commercially available pigments for visual artists to use and mix in various media such as oil, watercolor, acrylic, gouache, and pastel. A common approach is to use just a limited palette of primary pigments [20] often between four and eight that can be physically mixed to any color that the artist desires in the final work. Contemporary classical realists have often advocated that a limited palette of white, red, yellow, and black pigment often described as the "Zorn palette" is sufficient for compelling work. The human eye normally contains only three types of color photoreceptors, known as long-wavelength L , medium-wavelength M , and short-wavelength S cone cells. These photoreceptor types respond to different degrees across visible electromagnetic spectrum. The LMS primaries are complete since every visible color can be mapped to a triplet specifying the coordinates in LMS color space. Normalized cone spectral sensitivity curves The L, M and S response curves cone fundamentals were deduced from color matching functions obtained from controlled color matching experiments e. Any color space primaries which can be mapped to physiologically relevant LMS primaries by a linear transformation are necessarily either

imaginary or incomplete or both. The color-matching context is always three dimensional since LMS space is three dimensional but more general color appearance models like CIECAM02 describe color in six dimensions and can be used to predict how colors appear under different viewing conditions. Thus for trichromats like humans, we use three or more primaries for most general purposes. Adding a reasonable choice of third primary can drastically increase the available gamut, while adding a fourth or fifth may increase the gamut but typically not by as much. Most placental mammals other than primates have only two types of color photoreceptor and are therefore dichromats , so it is possible that certain combinations of just two primaries might cover some significant gamut relative to the range of their color perception. Meanwhile, birds and marsupials have four color photoreceptors in their eyes, and hence are tetrachromats. There is one scholarly report of a functional human tetrachromat. Measuring functional spectral discrimination in non-human animals is challenging due to the difficulty in performing psychophysical experiments on creatures with limited behavioral repertoires who cannot respond using language. Limitations in the discriminative ability of shrimp having twelve distinct color photoreceptors have demonstrated that having more cell types in itself need not always correlate with better functional color vision. Opponent process The opponent process is a color theory that states that the human visual system interprets information about color by processing signals from cones and rods in an antagonistic manner. The theory states that every color can be described as a mix along the three axes of red vs. The six colors from the pairs might be called "psychological primary colors", because any other color could be described in terms of some combination of these pairs. Although there is a great deal of evidence for opponency in the form of neural mechanisms, [27] there is currently no clear mapping of the psychological primaries to neural substrates. The Natural Color System is also directly inspired by the psychological primaries. Scholars and scientists engaged in debate over which hues best describe the primary color sensations of the eye. Hermann von Helmholtz proposed "a slightly purplish red, a vegetation-green, slightly yellowish, and an ultramarine-blue" as a trio.

Chapter 2 : A Little C Primer - Wikibooks, open books for an open world

COMPUTER NETWORKING PRIMER 3 A network also enables you to save money on software. Instead of buying separate copies of the same application for various machines, you can purchase one copy with.

Converting to and from higher-voltage logic Some situations require interfacing to control circuitry that works on 12V or other voltage substancially higher than the 5V or 3. I have, many times, used for example , , , and analog switches to control 12V analog circuits mostly audio , and these ICs needed the higher-voltage logic. Consider the common and cheap pin LM This diagram shows going from TTL-level logic to 12V logic levels: If you take the top of R1 to a different voltage, you will need to change the resistor value. If this goes to a separate board that only has 12V or other high voltage available and not 5V, you can of course take R1 to that voltage, and calculate the adjusted value. For going the other direction, ie, higher voltage to 5V logic, you can do something like this: It does not have to be 12V of course. The comparator is made to have its output hit the rails and recover more quickly than an op amp of similar price can. Bypass the reference voltage input to ground with another. Do not divide the signal input with resistors, since along with the input capacitance and other stray capacitance, you would get an RxC that would slow it down. If you really must divide it down, see the forum topic at Mixed Voltage Systems: Interfacing 5V and 3. Keep the pull-up resistor at around 6. There may be a temptation to increase the value to save power; but that will slow it down. Minimize the capacitive load on the output. Note that the LP is the lower-power version, but it is also much slower than the LM This requires logic-level translation, hence the LM; but although it worked for many years on many different PIC variations I programmed, there was trouble sometimes with the PIC16F72 and PIC16F74 which turned out to be because the slew rate on the clock line coming out of an LM section was not fast enough. The solution was to add a pair of 74HC14 Schmitt-trigger inverter sections in series to improve the rise time. For going between 5V and 3. Additionally, after I wrote the above, someone alerted me to the 74LS06 inverting and 74LS07 non-inverting open-collector hex buffers which are much faster than the and have a maximum output voltage of 30V but whose inputs do make for a greater load and are not available in CMOS. CMOS will be able to pull it up higher but the overdrive is nowhere near enough to hurt anything. If you need a lot of these, you might do better to use an IC like the ULN which has 8 Darlington drivers in an pin package with integral resistors and protection diodes. That would replace 24 discrete components, three for each circuit above, times eight. The configuration shown above however allows the output voltage to come nearly all the way to ground if the load is light enough and the bias current strong enough for the chosen transistor. High-voltage shift registers If you have a lot of higher-voltage e. Again, the shift-register chains can be quite long. Note that there is no "74" in front of the numbers in this case. In the automated test equipment I designed, built, and programmed in approximately , I drove approximately 75 relays by way of Allegro Microsystems UCNA I think it was made by Sprague at the time 8-bit, serial-in, parallel-out pin shift registers which are rated for 50V, mA, made specifically for driving relays and other heavyish loads. Wikipedia has quite a write-up on it, and Philips has a seminar-style overview on it. It is a synchronous-serial interface that requires only two wires; a bi-directional data line and a clock line. Each transaction is begun with a "start" condition and then the address of the target device. More on that later. There is now another possibility for even faster twiddling of these bits, using illegal op codes of the 65c Bit 7 along with bit 6 are easily tested with the BIT instruction without affecting or needing any processor registers. Otherwise you would need to read the port into the accumulator and use AND to test the desired bit. You can shut it down when not beeping by setting PB7 high. In that case, using the bits I assigned above to reduce the number of instructions may not give any real advantage; so you might as well assign the bits any way you like. I will address SPI later. I already have generic code to bit-bang SPI posted here. Next you might be wondering about the pull-up resistors. Devices can only pull it down, not up. Ok, so how do you have the VIA only pull down and not up? The interesting thing about that is that since a "1" in the data-direction register bit means "output" and a "0" means "input", pulling it down requires setting it to "1", and letting it float up means setting it to "0", instead of vice-versa. This is one place where the is far more efficient codewise than the or which do not allow a light-footed method of quickly going from accessing the

ports to accessing the data-direction registers and vice-versa. Each class of device has an address group assigned to a few bits in the address byte, and then the device might also have a few extra pins that you can connect to Vcc or ground to provide more bits in the address byte and allow several devices of the same class to be on the bus at the same time. If you want to make module to plug in, I would suggest the I2C-6 connector standard we devised on the The page has diagrams and photos. Various devices will have their own detailed operation which will be given in their data sheets; but there will be basic things that they all share. Interfacing to SPI and Microwire Another popular synchronous-serial interface is SPI pronounced "ess-pee-eye" which stands for "serial peripheral interface. SPI was named by Motorola who rounded up various synchronous-serial protocols that were already in use and asigned the mode numbers and tried to make it all more understandable. Wikipedia has quite a write-up on SPI. Bit-banging with a VIA as shown below will not operate SPI anywhere near its maximum speed, but it still enables us to help ourselves to hundreds if not thousands of great ICs on the market, made by dozens of manufacturers. For example, you can set or clear an output bit in a single clock cycle without affecting the other bits in the port! SPI normally involves four wires for one SPI device or slave , plus an extra one for each additional device. The SPI modes used by the different slaves on the bus do not need to match, as only one slave will be enabled at a time unless you daisychain, something which will not be addressed here. The arrows show signal direction. Since normally no line is bi-directional, it makes it easy to convert logic voltage levels if necessary. This is shown below, along with the entire 65SIB Half of it is just to feed the annunciator LEDs. Solid arrows on connections except to V R , the reference voltage show signal direction. Explanation is given in the 65SIB specification. Intelligent 65SIB devices can use it, but it will have no effect on non-intelligent devices, so in that case you could use it for something else on them if desired. I have ones here that do that. The pull-up resistors as mentioned in the notes for the simpler diagram further up are included, to prevent ambiguous states before the VIA is initialized in software. I used an LM open-collector quad comparator mainly because of the IRQ output; but there are different ways you could handle the comparator functions here. They do not need to be super fast. The LM does not do well with inputs that are less than 1. For other explanations on 65SIB, see the specification. See the spec here. The first SPI module available is a tiny flash memory one, shown here. Keypads and keyboards There are many ways to implement a keypad or keyboard. Going to the next step, you can have several buttons connected to output bits, with a common connection to a pulled-up input bit, like this: Note that the output bits can be connected to other things as well. A high input bit means the key that is on the low output is not being pressed. Make the software cycle through them. More on software considerations in a minute. The five-key keypad is what I have on my workbench computer , and the five VIA output bits feeding the keys also go to the LCD and my printer interface. I do the software development on a DOS PC and send code over the RS line; but the keypad is convenient for various things once a piece of code is set to running. The keys can be used for anything you determine in your software, but the functions I usually give mine are:

Chapter 3 : Quantum Computing Primer | D-Wave Systems

Get this from a library! The computer primer: for gifted beginners. [Ann Cavanaugh] -- An introduction to the world of computers, including their history, types, uses, parts, flowcharting, programming in BASIC, ethics, and careers.

Netbooks are the latest version of the portable computer. Similar to laptop computers in many ways except that Netbook computers are smaller, lighter, less expensive and less powerful. They are great for portability but less functional for computing intensive tasks. This Netbook Computer Primer examines just what a Netbook computer is and explores the potential benefits and drawbacks of this new breed of portable computer. What is a Netbook Computer? Based on current technology a Netbook computer is an ultraportable computer: Netbook computers are designed for less computing intensive tasks such as receiving and sending email and accessing the Internet. Netbooks are basic computing devices that are suited for basic computing tasks. Based on size and weight a Netbook fits somewhere in between the smartphone and the laptop computer. Key factors that differentiate a Netbook computer from other portable computers are: Upgrading to 2GB if needed will also add to the cost. Most Netbooks weigh in at around 2. A smaller screen means a lighter screen and a smaller and lighter battery to power the smaller screen. The weight of most Netbooks falls in the 2 â€" 3 pounds range however adding a larger, extended life battery can add as much as half a pound to the overall weight. While these CPUs provide less computing power they also require much less electrical power resulting in smaller, lighter weight batteries and cooler running Netbooks. Intel released its latest versions of the Atom CPU in early External drives can be connected through USB ports for loading software or transferring data to another computer. External optical drives are not included in the base cost of most Netbook computers. There are also a number of trade-offs with the reduced size and weight. The layout and arrangement of the keys can also vary. Netbooks that have inch screens tend to offer the larger keyboards. Well worth a test drive to determine if the smaller keyboard is a good fit for you. Computing power â€" As explained earlier a Netbook computer is not designed for computing intensive tasks such as editing videos or large photos or playing modern video games. Loading programs and saving documents may also be a bit slower than a laptop. After the space required for Windows and your program files these smaller capacity drives may not leave sufficient space for all your other files in the long term. Especially if you are considering copying your 30GB music and photo library onto the computer. It is relatively easy to connect an external drive via USB but it will cost extra. Battery life â€" These rating can vary significantly, ranging from 2. Many netbook computer manufacturers also offer an extended life battery as an extra cost option. The higher amount will provide better performance but is an added cost. Mouse buttons and touchpad â€" The smaller amount of keyboard real estate can also impact the mouse buttons and the touchpad. As with laptops the configuration of these important elements can vary. Another area where a test drive will be beneficial. Netbook computers provide some very interesting options. As the technology continues to evolve in this hot product area the capabilities are only going to get better and some of the trade offs, such as performance, may decrease significantly. The popularity of Netbooks is also putting additional pressure the price and performance of conventional, fully sized ultralight laptops. The distinctions between these two categories will probably continue to blur. For students and parents looking for a portable computer for school be sure to see the post on 7 Reasons Netbook Computers are Great for Students For many people the main decision point between a Netbook computer and a laptop will be size and weight. If you are looking for a very small footprint, very lightweight basic computing device you should seriously consider one these. Hard to Beat the Price of a Net book Computer Netbook computers are a relatively new market and in a significant state of flux. In an era of high performance quad-core CPUs and terabyte hard disk drives many people have been surprised by the popularity of these relatively low horsepower devices. A key factor that makes a Netbook computer so interesting is their low price.

Chapter 4 : A Translator's Tool Box

To provide an introduction to the content of the Computer Systems, Fundamentals of Computing, and Programming modules, from the MSc CS and MSc DS programmes. Syllabus The syllabus is provisional and is adapted dynamically to cope with the varying requirements of the student cohort.

As recently as 10 years ago, most clinician orders were handwritten. The vast majority of hospitals and most outpatient practices now use some form of CPOE. CPOE systems were originally developed to improve the safety of medication orders, but modern systems now allow electronic ordering of tests, procedures, and consultations as well. The widespread implementation of CPOE has benefited clinicians and patients, but it also vividly illustrates the risks and unintended consequences of digitizing a fundamental health care process. The process of prescribing and administering a medication involves several steps, each of which has vulnerabilities that are addressedâ€"to greater or lesser degreesâ€"by CPOE: These errors had a variety of causes, including poor handwriting, ambiguous abbreviations, or simple lack of knowledge on the part of the ordering clinician. A CPOE system can prevent errors at the ordering and transcribing stages by at a minimum ensuring standardized, legible, and complete orders. CPOE systems are generally paired with some form of clinical decision support system CDSS , which can help prevent errors at the medication ordering and dispensing stages and can improve safety of other types of orders as well. A typical CDSS suggests default values for drug doses, routes of administration, and frequency and may offer more sophisticated drug safety features, such as checking for drug allergies or drugâ€"drug or even drugâ€"laboratory e. The most sophisticated CDSSs prevent not only errors of commission e. CDSSs are also increasingly being deployed to address overuse â€"for example, a systematic review of CPOE for radiologic studies found that CDSS can improve adherence to guidelines for diagnostic imaging and reduce overall test usage. Evidence of Effectiveness CPOE offers numerous advantages over traditional paper-based order-writing systems. Examples of these advantages include: Supported by early evidence , the proposed benefits of CPOE served as a core part of the argument for federal funding to support the widespread implementation of CPOE. These proposed benefits have been borne out to some extent, principally with regard to improving medication safety. Specifically, CPOE appears to be effective at preventing medication prescribing errors. Studies of e-prescribing systemsâ€"CPOE systems used primarily in outpatient practices that allow direct transmittal of prescriptions to pharmaciesâ€"have also found similar effectiveness at preventing outpatient prescribing errors. The effect of CPOE on clinical adverse drug event rates is less clear. Other reviews have found that CPOE does not reliably prevent patient harm, and high rates of adverse drug events persist in some hospitals with entirely computerized order entry systems. One interpretation of these results is that clinical decision support is the key intervention in reducing errors, and that, in the absence of CDSS, CPOE may prevent mostly clinically inconsequential errors. However, usability testing has demonstrated that CPOE systems with clinical decision support still allow unsafe orders to be entered and processed, and that clinicians can bypass safety steps with little difficulty. Another interpretation is that a significant proportion of medication errors occur at the dispensing and administration stages, and CPOE may not prevent these errors. Promising error reduction strategies in the setting of dispensing and administration include involving unit-based pharmacists and using barcode medication administration systems. Yet even as CPOE improves some aspects of patient safety, there is growing recognition that it can also lead to new safety concernsâ€"particularly if the system is poorly designed. Implementation Issues and Workflow Impact of CPOE The implementation of CPOE has proven to be a complex process, and early users experienced high-profile failures or safety hazards that in some cases led to abandonment of the system. A great deal of research has characterized the types of unintended consequences and disruptions to clinician workflow that result from CPOE implementation. One study conducted after implementation of a commercial CPOE system found that the system required clinicians to perform many new tasks, increasing cognitive load and decreasing efficiency, and therefore raising the potential for error. In that study, although overall prescribing errors decreased, problems related to the CPOE system itself accounted for almost half of prescribing errors after implementation. Other studies have shown

that users often use workarounds to bypass safety features. In many cases, these workarounds represent reasonable adaptations due to problems with the design and usability of CPOE systems. As detailed in a Food and Drug Administration white paper summarized here , current CPOE systems have fundamental problems such as confusing displays, use of nonstandard terminology, and lack of standards for alerts and warnings. The authors call for integration of human factors engineering principles, including real-world usability and vulnerability testing, in order to achieve the safety potential of CPOE. Types of unintended consequences related to computerized provider order entry. J Am Med Inform Assoc. The integration of clinical decision support into CPOE systems also requires careful planning. Decision support alerts can prevent harmful drugâ€"drug interactions and promote use of evidence-based tests and treatments. However, excessive and nonspecific warnings can lead to alert fatigueâ€"whereby users ignore even critical warnings. Alert fatigue is now a recognized safety threat in itself and is discussed in detail in a related Patient Safety Primer. Alert fatigue likely explains why CDSSs appear to result in only modest improvements in adherence to recommended care and may fail to prevent errors. Recent research has focused on tailoring alerts to maximize safety while avoiding alert fatigue, but the informatics field has not yet developed standard approaches to achieve this balance. Reasons provided by prescribers when overriding drug-drug interaction alerts. Am J Manag Care. Effective CPOE implementation requires considerable investment of time and resources as well as commitment from both CPOE vendors and organizational leadership to ensuring safe integration of the technology with existing workflows. Adoption in the outpatient setting is also rapidly increasing, and as of the end of , more than half of office practices had adopted electronic prescribing the major form of CPOE in the outpatient setting.

Chapter 5 : Primary color - Wikipedia

Working on a computer involves creating and manipulating files. The data and the programs cannot be continuously stored in a computer's memory: they have to be stored in files that can be read by the computer when needed.

Media Resources Quantum Computing Primer This tutorial is intended to introduce the concepts and terminology used in Quantum Computing, to provide an overview of what a Quantum Computer is, and why you would want to program one. The material here is written using very high level concepts and is designed to be accessible to both technical and non-technical audiences. Some background in physics, mathematics and programming is useful to help understand the concepts presented in this document, although this is not a requirement. What you will learn By following through the material in this primer, you will learn: We take modern digital computers and their ability to perform a multitude of different applications for granted. Our desktop PCs, laptops and smart phones can run spreadsheets, stream live video, allow us to chat with people on the other side of the world, and immerse us in realistic 3D environments. But at their core, all digital computers have something in common. They all perform simple arithmetic operations. Their power comes from the immense speed at which they are able to do this. Computers perform billions of operations per second. These operations are performed so quickly that they allow us to run very complex high level applications. Conventional digital computing can be summed up by the diagram shown in figure 1. Dataflow in a conventional computer Although there are many tasks that conventional computers are very good at, there are still some areas where calculations seem to be exceedingly difficult. Examples of these areas are: Image recognition, natural language getting a computer to understand what we mean if we speak to it using our own language rather than a programming language , and tasks where a computer must learn from experience to become better at a particular task. Even though there has been much effort and research poured into this field over the past few decades, our progress in this area has been slow and the prototypes that we do have working usually require very large supercomputers to run them, consuming a vast quantities of space and power. We can ask the question: Is there a different way of designing computing systems, from the ground up? If we could start again from scratch and do something completely different, to be better at these tasks that conventional computers find hard, how would we go about building a new type of computer? With quantum computing, everything changes. The physics that we use to understand bits of information and the devices that manipulate them are totally different. The way in which we build such devices is different, requiring new materials, new design rules and new processor architectures. Finally, the way we program these systems is entirely different. This document will explore the first of these issues, how replacing the conventional bit 0 or 1 with a new type of information - the qubit - can change the way we think about computing. The light switch game involves trying to find the best settings for a bunch of switches. This gives us a number. The objective of the game is to set the switches to get the lowest number. You can try this game. Working out the answer for a particular "guess" at the switch settings We find that if we set all the switches with positive biases to OFF and all the switches with negative biases to ON and add up the result then we get the lowest overall value. I can give you as many switches as I want with many different bias values and you just look at each one in turn and flip it either ON or OFF accordingly. So now imagine that many of the pairs of switches have an additional rule, one which involves considering PAIRS of switches in addition to just individual switches Making the game harder by adding additional terms that depend on the settings of pairs of switches. But now it is much, much harder to decide whether a switch should be ON or OFF, because its neighbors affect it. With a complex web of switches having many neighbors, it quickly becomes very frustrating to try and find the right combination to give you the lowest value overall. But as you add more and more switches, the number of possible ways that the switches can be set grows exponentially: The exponential problem with the light switch game. In fact it is even difficult for our most powerful supercomputers. Being able to store all those possible configurations in memory, and moving them around inside conventional processors to calculate if our guess is right takes a very, very long time. Quantum mechanics can give us a helping hand with this problem. The fundamental power of a quantum computer comes from the idea that you can put bits of information into a

superposition of states. You can think of this as being a situation where the qubit has not yet decided which state it wants to be in. A network of connected quantum bits in superposition. The answer is in there somewhere! But that is OK, because quantum mechanics is going to find it for us. The computer begins with the bits in superposition, ends with them behaving as regular classical bits, and finds the answer along the way. You start with the system in its quantum superposition as described above, and you slowly adjust the quantum computer to turn off the quantum superposition effect. As this operation is performed, the switches slowly drop out of their superposition state and choose a classical state, either ON or OFF. The quantum mechanics working inside the computer helps the light switches settle into the right states to give the lowest overall value when you add them all up at the end. So we see that the quantum computer allows us to minimize expressions such as the one considered here: The concept of finding a good configuration of binary variables switches in this way lies at the heart of many problems that are encountered in everyday applications. A few are shown in figure below. Recall Figure 1, where bit strings in were transformed into other bits strings via the application of a logic program. Instead of that, we now have a resource where bits can be undecided, so the computation is performed in a fundamentally different way, as shown in Figure What is an Energy Program? It is just those h and J numbers - the bias settings - that we introduced earlier. Well, now we see where they come from - they are the definition of the problem you are trying to solve. Crafting an energy program as a series of h and J values - to encode a real world problem that you care about - is exceedingly hard and time-consuming. It would be the equivalent of programming your desktop PC by sending in machine code to the microprocessors inside! Luckily, there is a better way to program quantum computers by using a quantum compiler. This process is explained in more detail in the Programming with D-Wave white paper. It is a sub-branch of the field of artificial intelligence. Most of the code we write is fairly static - that is, given new data it will perform the same computation over and over again and make the same errors. Using machine learning we can design programs which modify their own code and therefore learn new ways to handle pieces of data that they have never seen before. For example, imagine if a computer was asked to classify an object based on several images of similar objects you had shown it in the past. This task is very difficult for conventional computing architectures, which are designed to follow very strict logical reasoning. How can we use a quantum computer to implement learning, for example, if we want the system to recognize objects? Writing an energy program for this task would be very difficult, even using a quantum compiler, as we do not know in detail how to capture the essence of objects that the system must recognize. Luckily there is a way around this problem, as there is a mode in which the quantum computer can tweak its own energy program in response to new pieces of incoming data. This allows the machine to make a good guess at what an object might be, even if it has never seen a particular instance of it before. The following section gives an overview of this process. An example in shown in Figure Here the idea is to try to get the computer to learn the difference between images of different types of fruit. In order to do this, we present images or rather, a numeric representation of those images to the system illustrating many different examples of apples, raspberries and melons. The system must find an energy program shown as a question mark as we do not know it at the beginning so that when an image is shown to the system, it gets the labels correct each time. If it gets many examples wrong, the algorithm knows that it must change its energy program. Teaching the quantum chip by allowing it to write its own energy program. The system tweaks the energy program until it labels all the examples that you show it correctly. At first the system chooses an energy program remember that it is just a bunch of H and J values at random. In machine learning terminology this is known as a supervised learning algorithm because we are showing the computer examples of images and telling it what the correct labels should be in order to help it learn. There are other types of learning algorithms supported by the system, even ones that can be used if labeled data is not available. Some of these might be the answer that you are looking for, and some might not. At first this sounds like a bad thing, as a computer that returns a different answer when you ask it the same question sounds like a bug! However, in the quantum computer, this returning of multiple answers can give us important information about the confidence level of the computer. However, if it returns the answer apple 50 times and raspberry 50 times, what this means is that the computer is uncertain about the image you are showing it. And if you had shown it an image with apples AND

raspberries in, it would be perfectly correct! This uncertainty can be very powerful when you are designing systems which are able to make complex decisions and learn about the world.

Chapter 6 : Primer | Stanford Computer Science

*The Computer Primer for Gifted Beginners [Ann Cavanaugh] on calendrierdelascience.com *FREE* shipping on qualifying offers. An introduction to the world of computers, including their history, types, uses, parts, flowcharting.*

Timeline of notable computer viruses and worms Early academic work on self-replicating programs[edit] The first academic work on the theory of self-replicating computer programs [18] was done in by John von Neumann who gave lectures at the University of Illinois about the "Theory and Organization of Complicated Automata ". The work of von Neumann was later published as the "Theory of self-reproducing automata". In his essay von Neumann described how a computer program could be designed to reproduce itself. The Reaper program was created to delete Creeper. On its 50th use the Elk Cloner virus would be activated, infecting the personal computer and displaying a short poem beginning "Elk Cloner: The program with a personality. In , Fred Cohen published a demonstration that there is no algorithm that can perfectly detect all possible viruses. However, antivirus professionals do not accept the concept of "benevolent viruses", as any desired function can be implemented without involving a virus automatic compression, for instance, is available under the Windows operating system at the choice of the user. Any virus will by definition make unauthorised changes to a computer, which is undesirable even if no damage is done or intended. Gunn under the title "Use of virus functions to provide a virtual APL interpreter under user control" in A few years later, in February , Australian hackers from the virus-writing crew VLAD created the Bizatch virus also known as "Boza" virus , which was the first known virus to target Windows In late the encrypted, memory-resident stealth virus Win Cabanas was releasedâ€”the first known virus that targeted Windows NT it was also able to infect Windows 3. The first one to appear on the Commodore Amiga was a boot sector virus called SCA virus , which was detected in November Users would be required to click on a link to activate the virus, which would then send an email containing user data to an anonymous email address , which was later found to be owned by Larose. Data sent would contain items such as user IP address and email addresses, contacts, website browsing history, and commonly used phrases. In , larger websites used part of the Win Operations and functions[edit] Parts[edit] A viable computer virus must contain a search routine , which locates new files or new disks which are worthwhile targets for infection. Secondly, every computer virus must contain a routine to copy itself into the program which the search routine locates. A virus typically has a search routine, which locates new files or new disks for infection. Payload activity might be noticeable e. This life cycle can be divided into four phases: Dormant phase[edit] The virus program is idle during this stage. The virus will eventually be activated by the "trigger" which states which event will execute the virus, such as a date, the presence of another program or file, the capacity of the disk exceeding some limit or the user taking a certain action e. Not all viruses have this stage. The virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often "morph" or change to evade detection by IT professionals and anti-virus software. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase. The triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself. It can be destructive such as deleting files on disk, crashing the system, or corrupting files or relatively harmless such as popping up humorous or political messages on screen. Resident viruses overwrite interrupt handling code or other functions , and when the operating system attempts to access the target file or disk sector, the virus code intercepts the request and redirects the control flow to the replication module, infecting the target. In contrast, a non-memory-resident virus or "non-resident virus" , when executed, scans the disk for targets, infects them, and then exits i. This is one of the reasons that it is dangerous to open unexpected or suspicious attachments in e-mails. Boot sector viruses[edit] Email virus[edit] Email virus â€” A virus that intentionally, rather than accidentally, uses the email system to spread. While virus infected files may be accidentally sent as email attachments , email viruses are aware of email system functions. Some old viruses, especially on the DOS platform, make sure that the "last modified" date of a host file stays the same when the file is infected by the virus. This approach does not fool antivirus software , however, especially those which maintain and date

cyclic redundancy checks on file changes. They accomplish this by overwriting unused areas of executable files. These are called cavity viruses. Because those files have many empty gaps, the virus, which was 1 KB in length, did not add to the size of the file. In the s, as computers and operating systems grow larger and more complex, old hiding techniques need to be updated or replaced. Defending a computer against viruses may demand that a file system migrate towards detailed and explicit permission for every kind of file access. This leaves antivirus software little alternative but to send a "read" request to Windows OS files that handle such requests. Some viruses trick antivirus software by intercepting its requests to the Operating system OS. A virus can hide by intercepting the request to read the infected file, handling the request itself, and returning an uninfected version of the file to the antivirus software. The interception can occur by code injection of the actual operating system files that would handle the read request. Thus, an antivirus software attempting to detect the virus will either not be given permission to read the infected file, or, the "read" request will be served with the uninfected version of the same file. Security software can then be used to check the dormant operating system files. Most security software relies on virus signatures, or they employ heuristics. Self-modifying code Most modern antivirus programs try to find virus-patterns inside ordinary programs by scanning them for so-called virus signatures. Such a virus "signature" is merely a sequence of bytes that an antivirus program looks for because it is known to be part of the virus. A better term would be "search strings ". Different antivirus programs will employ different search strings, and indeed different search methods, when identifying viruses. If a virus scanner finds such a pattern in a file, it will perform other checks to make sure that it has found the virus, and not merely a coincidental sequence in an innocent file, before it notifies the user that the file is infected. The user can then delete, or in some cases "clean" or "heal" the infected file. Some viruses employ techniques that make detection by means of signatures difficult but probably not impossible. These viruses modify their code on each infection. That is, each infected file contains a different variant of the virus. If the virus is encrypted with a different key for each infected file, the only part of the virus that remains constant is the decrypting module, which would for example be appended to the end. In this case, a virus scanner cannot directly detect the virus using signatures, but it can still detect the decrypting module, which still makes indirect detection of the virus possible. Since these would be symmetric keys, stored on the infected host, it is entirely possible to decrypt the final virus, but this is probably not required, since self-modifying code is such a rarity that it may be reason for virus scanners to at least "flag" the file as suspicious. This is called cryptovirology. At said times, the executable will decrypt the virus and execute its hidden runtimes , infecting the computer and sometimes disabling the antivirus software. Just like regular encrypted viruses, a polymorphic virus infects files with an encrypted copy of itself, which is decoded by a decryption module. In the case of polymorphic viruses, however, this decryption module is also modified on each infection. A well-written polymorphic virus therefore has no parts which remain identical between infections, making it very difficult to detect directly using "signatures". To enable polymorphic code, the virus has to have a polymorphic engine also called "mutating engine" or " mutation engine" somewhere in its encrypted body. See polymorphic code for technical detail on how such engines operate. For example, a virus can be programmed to mutate only slightly over time, or it can be programmed to refrain from mutating when it infects a file on a computer that already contains copies of the virus. The advantage of using such slow polymorphic code is that it makes it more difficult for antivirus professionals and investigators to obtain representative samples of the virus, because "bait" files that are infected in one run will typically contain identical or similar samples of the virus. This will make it more likely that the detection by the virus scanner will be unreliable, and that some instances of the virus may be able to avoid detection. Metamorphic code[edit] To avoid being detected by emulation, some viruses rewrite themselves completely each time they are to infect new executables. Viruses that utilize this technique are said to be in metamorphic code. To enable metamorphism, a "metamorphic engine" is needed. A metamorphic virus is usually very large and complex. Software development strategies that produce large numbers of "bugs" will generally also produce potential exploitable "holes" or "entrances" for the virus. Social engineering and poor security practices[edit] In order to replicate itself, a virus must be permitted to execute code and write to memory. For this reason, many viruses attach themselves to executable files that may be part of legitimate programs see code injection. This

makes it possible to create a file that is of a different type than it appears to the user. For example, an executable may be created and named "picture. Many Windows users are running the same set of applications, enabling viruses to rapidly spread among Microsoft Windows systems by targeting the same exploits on large numbers of hosts. This difference has continued partly due to the widespread use of administrator accounts in contemporary versions like Windows XP. In , researchers created and released a virus for Linuxâ€”known as " Bliss ". Unlike Windows users, most Unix users do not log in as an administrator, or "root user" , except to install or configure software; as a result, even if a user ran the virus, it could not harm their operating system. The Bliss virus never became widespread, and remains chiefly a research curiosity. Its creator later posted the source code to Usenet , allowing researchers to see how it worked.

Chapter 7 : Spelling Practice with Dolch Sight Words | ABCya!

Second, it provides a primer of the most frequently used weapon for addressing computer fraud, the Computer Fraud and Abuse Act (CFAA). This will allow the litigator to be better prepared for the inevitable computer fraud battles that lie ahead, as well as to advise clients on how to avoid violating these laws, and, when necessary, use them.

Introduction to filesystems Working on a computer involves creating and manipulating files. The various methods for handling files are called filesystems. However, with a bit of information, you can: Let other people read your files. Let some other people read some of your files, and not the others. Get to know the basics of how filesystems actually work. Local filesystems handle files that are stored directly on the computer you are using. These files are stored on a local hard disk, sometimes called a hard drive. In the Computer Science Department, local drives are used to hold programs and data needed to recover from reboots, temporary user data scratch space , and most commonly used application programs. Distributed filesystems do not store files directly on the local computer, but rather in a central storage location that is accessible to a network of computers. This storage space, called a fileserver, is a computer with an extra large hard drive. All personal files are stored on fileservers and are accessible from any CS workstation. However, since all the files have to pass through the network to get to your workstation, access is a little bit slower than for local files. NFS is a distributed filesystem. Its files are stored on fileservers; in order to access a files, you have to mount it to your computer. Local and NFS file and directory permissions File permissions regulate file access. They allow you to control who can read, write, or execute any files you may own. File permissions in a local filesystem and in NFS are similar, and specified for each file and directory. The three basic permissions are: Read permission on a directory implies the ability to list all the files in the directory. Files that are not programs should not be given the execute permission. For directories, execute permission allows you to enter the directory i. Under both local filesystems and NFS, permissions exist separately for user, group, and others. User u permissions apply to the owner of the file. Group g permissions apply to all members of the group associated with the file. Permissions for others o apply to anyone else. The default owner of any file you create will be you. The group will be inherited from its parent directory. The permissions, owner and group associated with a file or directory can be checked by looking at the output of ls -lg. The permissions are listed in the first field of the output. If the first character is d, the entry is a directory; the character - or the letter f signifies a normal file. The next three characters signify the user permissions: The following three characters represent the group permissions in the same way, and the last three represent others. The owner of the file is listed in the third field of the output; the fourth field lists the group associated with the file. So if a file notes looked like this: Note that the username root signifies the superuser. Anyone with the superuser privileges implicitly has read and write permissions, as well as the ability to change permissions, on all the files on the local disks. The permissions can be changed by using the chmod command. Similarly, permissions are withdrawn by using chmod who-permissions. You can change the group associated with a file by chgrp groupname file s. For more information on these two commands, read the corresponding manual pages via man chown and man chgrp on any CS workstation. You may also notice new subdirectories in your home directory such as. A cell constitutes a separate administrative domain of authority. Each cell keeps its own list of users, groups, and system administrators. That means that a user from one cell might not exist in another cell. In that case, they will only be able to access the files in directories that have the appropriate permission set to system: An example of a cell is the cs. Each cell is made up of volumes. A volume is a collection of files and directories that are grouped together and given a name. Your home directory is a volume, the volume user. Once created, each volume can as a unit be moved from one server to another, backed up, replicated or destroyed. Files and directories can be created, modified or deleted only in an existing volume. These volumes act like directories, and may in turn contain the mountpoints to other volumes. Because of the way AFS works, you do not have to explicitly attach any volume or filesystem that is on AFS in order to have access to it. All you need, in order to access a file, is the pathname of the file. Volumes that are in other cells outside the cs. To do this you need to get tokens for the cell, which are analogous to separate Kerberos tickets for

individual NFS fileservers. The most common way to get tokens is to use the command klog, described later on this page. Backup and read-only volumes There are three types of volumes: A read-write volume is a regular volume that can be read and written-just as the name implies. A read-write volume may have associated with it zero, one, or many read-only volumes. Read-only volumes cannot be modified by normal users. They have special properties, the most important of which is that many copies of a read-only volume can exist at once. If an AFS mountpoint is read-only and a read-only volume exists with the right name, AFS just picks one read-only volume to read from. If that volume disappears or somehow becomes unreachable, AFS will start using another one without the user ever knowing the difference. Backup volumes are also special. There can be only one backup volume for a read-write volume. Read-only volumes cannot have backup volumes. In other words, a backup volume can be associated only with a read-write volume. A backup volume is a read-only copy of a read-write volume that actually shares the same disk space as the read-write volume. These volumes are often known as clones. When a volume is backed up, that volume initially takes a very small amount of space on disk. As the read-write volume and the backup volume get further out of synchronization, data is actually copied. The next time the volume is backed up, the old copied data is destroyed. That means that, once a volume has been backed up once, subsequent backups of the volume may actually reduce the total amount of disk spaced used. It is a mountpoint for the volume user. All your files are backed up nightly. If you want to retrieve a file that you have accidentally removed, all you have to do is cd to the appropriate directory within. In AFS, file permissions are specified for each directory, and apply to the directory and to all the files that this directory contains. They do not apply to the subdirectories of a directory, since the subdirectories have their own permissions; however, any newly created subdirectory will inherit the permissions of its parent directory. These directory permissions are flexible; they can be applied individually for each user. You can give Jim, Mary and Bill the permission to see the list of all files in your home directory, Valerie the permission to list and read them, and Tom the permission to list, read and write them. The list of all users that have permissions, along with their permissions, is called the access control list or ACL of the directory. ACL is often pronounced ackle. There are seven types of access that you can grant: It does not imply read access to the files. You must have lookup permission to use any other permission except administer. Read Read access on a directory implies permission to read the contents of all the files in a directory. This says nothing about the right to read files in its subdirectories. Write Write access on a directory grants permission to modify existing files and subdirectories within a directory, and to change permissions on the files in that directory. It implies neither insert nor delete access to the directory. Insert Insert access on a directory implies permission to create files or subdirectories in the directory. It does not imply the ability to modify the files once they are created, however. Insert access without write access is useful mainly for the case when you want to allow someone to create files or subdirectories in a given directory but not to modify files that are already there. Delete Delete access on a directory gives the ability to remove files or empty subdirectories from the directory. Like insert, delete access does not imply write access. Administer With administer access on a directory, it is possible to change the ACL of the directory. Administer access does not imply any other kind of access. As with all other rights, setting or resetting administer access on a directory only affects that particular directory.

Chapter 8 : Sight Word BINGO - Quickly Match Sounds to Words

This Netbook Computer Primer examines just what a Netbook computer is and explores the potential benefits and drawbacks of this new breed of portable computer. What is a Netbook Computer? Let's start at the beginning.

This is Primer 1 in a series of seven that will calmly introduce you to the very basics of HyperText Mark-up Language. I suggest you take the Primers one at a time over seven days. I say that because many people scoff at the notion that they can actually learn this new Internet format. Some of the smartest people I know love to proclaim themselves "Dummies" regarding every aspect of computers. You Can Do This! I am assuming, however, some computer knowledge. To continue with these Primers, you will need A computer obviously 2. If you look up at the title bar at the very top of your screen it will probably say the page title "Basic HTML: If you have those three things, you can write HTML with the best of them. Now here are a few questions you probably have: HTML does not use any specific platform. It works with simple text. More on that in a moment Must I be logged onto the Internet to do this? More specifically, will learning this throw my cost for on-line way up? You will write off-line. Do I need some sort of expensive program to help me write this? You will write using just what I outlined above. HTML is not a computer language. Allow me to repeat that in bold HTML is not a computer language! Let me break it down for you: Hyper is the opposite of linear. It used to be that computer programs had to move in a linear fashion. This before this, this before this, and so on. HTML does not hold to that pattern and allows the person viewing the World Wide Web page to go anywhere, any time they want. Text is what you will use. Real, honest to goodness English letters. Mark up is what you will do. You will write in plain English and then mark up what you wrote. More to come on that in the next Primer. Here, I want to tell you how you will go about the process. Now, some people who are already schooled in HTML are going to jump up and down and yell that you should be using an HTML assistant program because it makes it easier. Take my word for it, use the word processor for a week, then go to the assistant if you still want to use one. Keep this in mind: HTML documents must be text only. When you save an HTML document, you must save only the text, nothing else. The reason I am pushing NotePad, WordPad, and Simple Text is that they save in text-only format without your doing any additional work. They just do it. If so, read this next part carefully. The Word Processor When you write to the word processor you will need to follow a few steps: Write the page as you would any other document. Usually at the bottom, you find where you will be able to change the file format. Either one will work. It just helped me keep it all straight, but if you want to save right to your hard drive, do it. I only offer the floppy disc premise as a suggestion. You see, when you save your document in WORD, or some other word processor format other than text, you are saving much more than just the letters on the page. You just want the text. You must first give your document a name and then add a suffix to it. You give a name and then a suffix. Follow this format to name your document: If you have a PC not running Windows 95, you are limited to eight letters, however. For all HTML documents, you will add either ". I am looking to name a document I just wrote on a PC running Windows 3. I want to name the document "fred". Thus the document must be named "fred. Please notice the dot period before. And no quotation marks, I just put them in here to set the name apart. Why Do I Do That? All files used on the Web will follow the format of "name. Your browser allows for both suffixes. You see, HTML browsers can only read text. Look at your keyboard. There are in all read upper- and lowercase letters as two. Then try and open it in your browser. Go ahead and try it. Remember that if you are using Notepad, Wordpad, or Simple Text, the document will be saved as text with no extra prompting. Some browsers give you the dialogue box that allows you to find your document right away. When the dialogue box opens up, switch to the A: If you saved the file to your hard drive, get it from there. You might have to then click an OK button. The browser will do the rest. One More Thing You easily have enough to keep you occupied for the first day. What I mean is for you to look at the HTML document a person wrote to present the page you are looking at. Why Would I Do That? If you see a room layout you like, you will use the idea to help yourself. It was the long way around, believe me. When you find a page you like, click on VIEW at the top of the screen. The HTML document will appear on the screen. Try it with this page. You can do it by placing your pointer on the page, off of an image,

and clicking the right mouse button. MAC users should click and hold. A small menu should pop up. One of the items will allow you the ability to view the source.

Chapter 9 : Basic HTML: Introduction

A Computer Primer for Translators Version As any good craftsman can tell you, it's not enough to simply have the right tool for the jobâ€”you've also got to know just how to use it.