Basic electronics. Electricity is either AC for alternating current or DC for direct current. In the US, our AC is 120V 60 Hz. Hz stands for Hertz, and that is how many times per second the current field flips, or alternates. 60 Hz means it flips 60 times each second. When a generator produces electricity, it uses a magnet spinning inside a coil of two armatures made from copper wire wrapped around a core and connected to a load. As the magnet spins, each pole alternates between both coils. When it faces one way, that makes the current flow that way, and when it faces the other it switches. When generating DC electricity, the coil is spinning between two magnets. The coil transfers the electricity to the load using brushes which take the current from the ends of the coil. This way, the current doesn't change directions as it spins since the north and south poles of the magnet are fixed. We will be concentrating on DC electricity.

Voltage is measured in volts (V), current is measured in amps (A on circuits, I in formulas), and resistance is measured in Ohms (Ω on circuits, R in formulas). Electrons have a negative charge and are attracted to a positive charge. Current moves in the opposite direction. When you are tracing a circuit, current goes to the negative side. Resistance slows down this movement though a formula called Ohm's law, which is used to determine the amount of current, voltage, or resistance in a circuit. V=I*R, I=V/R, R=V/I. Transformers are used to increase and decrease voltages.

When we talk about conducting electricity, we know that metal tends to be a better conductor than other things, and some metal is better than others. This has to do with the number of free electrons in the outer valence, but you don't need to remember that, just that metal conducts and wood insulates. Gold is the best conductor, then silver, copper, and aluminum. As electricity travels through circuits, it loses some of the voltage and current through resistance. This happens when the electrons collide with the lattice of the material it is traveling across. For a long aluminum wire, the voltage drop is greater than when using a short gold wire.

Resistors are components that are made to have a set amount of resistance. They are made out of several materials: wire wound, ceramic film, carbon composition, metal film, metal oxide film, and foil. They are marked with a series of colored stripes or numbers for the amount of resistance. Resistors have no polarity, meaning current can travel both directions across a resistor and it will work the same.

Capacitors take current and store it for a period of time. They can be made of several materials: ceramics, electrolyte, tantalum, silver mica, or film. They work by having two conductors separated by a filler material that allows for a large potential difference of electrical

charge. Ceramic and film capacitors have no polarity while the others do. They are typically used to filter out AC noise from DC circuits, audio signal filtering, and smoothing out DC voltages.

Diodes are made to allow current to travel in one direction, so they have polarity. LEDs are light emitting diodes and are used for light sources such as flashlights. They are also used in reverse current protection circuits and rectifier circuits that change AC to DC. There are a few specific types of diodes such as zener diodes and Schottky diodes as well.

Transistors can either be an amplifier or a switch. They can take a small input and create a large output like in a microphone, or they can store a binary value. They are made by sandwiching different polarities of silicon together with an emitter, base, and a collector, and are called junction transistors. RAM consists of billions of transistors. They are used as logic gates and are how a CPU makes it calculations using just a few states: AND, OR, NOT, NOR, XOR, and NAND. A field effect transistor (FET) also has three legs, but they are called the source, gate, and drain. They are coated with metal oxide and called metal oxide semiconductor field effect transistors (MOSFET).

Safety is of the utmost importance when working with electronics. Some power boards I have worked with have over 10,000V and the capacitors hold a charge for a while. When you lay your hand on the pins, you will know it and it will not feel good. Since I have abridged this class to fit the curriculum, I have had to leave out some parts that you don't need to know, and one is how voltage is actually just a potential difference between two points. This matters now, because when you work with circuit boards you can damage them with static electricity. In order to prevent a static discharge, you ground yourself to the chassis of what you are working with before touching anything inside. It is also a good practice to wear a grounding strap and not work on carpet in your shop. Also, rings and jewelry that hangs down such as necklaces need to be removed. Wash your hands after handling circuit boards as they have chemicals that are dangerous if precautions aren't taken.

Computers. They all have some things in common but don't always look the same. A motherboard is a printed circuit board that holds all of the other items inside the computer. It has various connectors and places to plug cables and extension boards, also known as daughter boards, into. One of the main components is the CPU, central processing unit, which performs all of the computing, or processes all of the instructions. A CPU is made up of billions of transistors which hold data while performing the computations in areas called registers or caches, which are groups of transistors of different amounts of storage, usually 16, 32, and 64 bits for registers. Caches have much more storage, measured in MB. All of the

storage on a CPU is volatile, or loses the content when the power turns off.

Memory, also known as RAM, random access memory or ROM, read only memory. RAM is volatile and can be written to many times. ROM is not volatile but cannot be written to without special tools or software. The BIOS, basic input output system, or bootstrap program is stored in ROM. Video comes from the CPU, called on-board video or a separate card called a video card which plugs into the motherboard. Some CPUs don't come with on-board video, such as the Intel Xeon series, which is the same as an Intel i7 but made for servers which usually run in headless mode, meaning they don't have a GUI, graphical user interface. They have a video card for access when you need video. Audio comes from the audio section of the motherboard or sometimes from an audio card plugged into the motherboard. The network connection is usually on the NIC, network interface card, and is where the ethernet cable plugs into and looks like a phone jack. Phone jacks are known as RJ-11 while ethernet jacks are known as RJ-45. It can be integrated onto the motherboard or a separate card.

USB stands for universal serial bus and they are either integrated onto the motherboard (usually in the rear of the computer) or plug into the motherboard with a cable (usually in the front of the computer). Serial is the method of communication meaning that it sends data one bit at a time as opposed to parallel which sends multiple bytes at the same time. The DVD/CD drive plays data from DVDs or CDs. DVD drives can play CDs but CD drives cannot play DVDs. It usually connects to the motherboard with SATA, serial advanced technology attachment, connections. Older computers use an IDE, integrated drive electronics, connection. The hard drive is the mass storage device that holds the data in your computer when the power is turned off. They are traditionally made of spinning platters the data is written to (HDD) with a movable arm. Solid state drives (SSD) are now popular because they don't have any moving parts and are much faster than HDDs and they can be much smaller.

The power supply takes the 120V AC power from the outlet and converts it to the various DC voltages (12V+, 12V-, 5V+, 5V-, and 3.3V+) that the computer needs. All electronics use DC power and all household outlet power is AC, so before being used, the AC needs converted to DC with a power inverter which is then stepped down to the proper voltage. You should never touch certain places inside a power supply since the voltages of certain components can reach upwards of 10,000V and capacitors can hold voltages for a long time before draining.

Peripherals are various things a computer has that makes it usable including, keyboard and mouse - also known as human interface devices (HID), monitor, printer, and other specialty devices that are not common. Most printers now use USB connections as that is almost standard for just

about every device that connects to a computer but used to have parallel cables for this. Keyboards and mice now also use USB connections, but used to use specialty PS/2 connections which were round and green for the mouse and purple for the keyboard. Before that, they used DB9 or DB15 serial connections. That stands for D-subminiature, it resembles a letter D when you look at it, and the 9 or 15 is how many pins there are. Monitors use VGA (video graphics array), DVI (digital video interface), or HDMI (high definition media interface) connections. VGA is almost exclusively on-board video while DVI and HDMI are on the video cards. There are also display port adapters which look like an HDMI connector with one flat side which are becoming more common. These peripherals are also collectively known as input/output devices for obvious reasons.

The CPU, central processing unit, is the brain of the computer. They are described using architecture, such as x86 (32 bit), x86-64 (64 bit), ARM, and others. They are also identified by manufacturer, like Intel, AMD, Qualcom and others. Most CPUs use von Neumann architecture, which use the same path for data and program instructions. The CPU in your desktop and laptop is probably an Intel or AMD 64 bit. These are larger and consume more power than ARM chips, but are more powerful*. 32 bit versions are still around, but 64 bit is prevalent. This refers to the size of data it's bus can handle. CPUs only recognize 1’s and 0’s, aka binary language. The CPU in your phone is an ARM of some sort. They are smaller and more power efficient and run cooler.

In order for us to use a CPU, we need to change the binary to something we can use to move data around. The next step up is called assembly language and consists of commands of three or four letters, numbers, and commas (there are multiple versions of assembly, but these are common in most). The data is moved around the CPU in registers and caches of various sizes up to the limit, 32 or 64 bits. These instructions in turn tell the other data residing in RAM, known as the stack, where to go and what to do. This can be something like displaying the words on your screen, one small piece at a time. Most CPUs work at 2 or more GHz, which gives them over 2 billion clock cycles to work with every second.

Since I started at the bottom, I want to clarify. From the top, you write a program in C or Python, high level languages, and before they are executed, they are converted to assembly language by an assembler, either on the fly as it is running or all at once and loaded into memory. Then those instructions are executed and the CPU processes them, taking the various values from the registers and putting them into the proper places in RAM. This is where they get converted from assembly to binary. If you look at the contents of your RAM now, it would be a series of hexadecimal numbers, which represent their binary equivalent but are easier for us to read. That is what makes the computer work.

- With advances in technology, ARM chips are getting better, and may now be on par with x86 chips. Apple uses them in their new laptops.

When the operating system stores data on a hard drive it uses a file system so that it can keep track of them. Windows uses FAT32 (file allocation table), exFAT (extended file allocation table), and NTFS (new technology file system). Linux uses Ext2 (extended file system), Ext3, Ext4, Btrfs (B-tree file system), JFS (journaled file system), and ReiserFS. Apple uses APFS (apple file system), MacOS extended (also called HFS), FAT and exFAT. They all come in journaled, journaled-encrypted, case sensitive-journaled, and case sensitive-encrypted except FAT and exFAT.

When a hard drive is formatted, the file system keeps a map after dividing the disk into blocks. While they all work differently, they all do the same thing. The differences are in the details and are not that important, other than knowing which ones work on which file systems. Linux can read NTFS but not write to it, while Apple can read it and write to it with 3d party software. Windows can read or write to Ext and HFS with 3d party software. Another thing to remember is that when the map gets corrupted, the file system won't boot, but it is not necessarily unreadable.

Windows uses a registry to store databases of configuration files, also known as a hive. It has four parts called root keys: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE. HKEY_USERS, and HKEY_CURRENT_CONFIG. Each of these keys has various keys which may or may not have a sub key, until you get to the lowest key. They hold different types of data to tell the operating system all the information it needs to run properly. This is different from the user files where the working data resides. Windows labels the hard drive as C: by default, and the folder structure is written like: C:\Windows\System32. Almost all Windows components are not case sensitive so the file named System.txt is the same as system.txt.

Linux and Apple have a different type of file structure in that they don't have a registry and use a different naming convention, and although they are slightly different in the folders they contain, they work in the same manner so I will combine them for instructional purposes. The root folder is written with a / and all other folders come next and are written like:

/usr/bin. Linux and Apple are case sensitive, so System.txt and system.txt are two different files.

When the operating system stores data on a hard drive it uses a file system so that it can keep track of them. Windows uses FAT32 (file allocation table), exFAT (extended file allocation table), and NTFS (new technology file system). Linux uses Ext2 (extended file system), Ext3, Ext4, Btrfs (B-tree file system), JFS (journaled file system), and ReiserFS. Apple uses APFS (apple file system), MacOS extended (also called HFS), FAT and exFAT. They all come in journaled, journaled-encrypted, case sensitive-journaled, and case sensitive-encrypted except FAT and exFAT.

When a hard drive is formatted, the file system keeps a map after dividing the disk into blocks. While they all work differently, they all do the same thing. The differences are in the details and are not that important, other than knowing which ones work on which file systems. Linux can read NTFS but not write to it, while Apple can read it and write to it with 3d party software. Windows can read or write to Ext and HFS with 3d party software. Another thing to remember is that when the map gets corrupted, the file system won't boot, but it is not necessarily unreadable.

Windows uses a registry to store databases of configuration files, also known as a hive. It has four parts called root keys: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE. HKEY_USERS, and HKEY_CURRENT_CONFIG. Each of these keys has various keys which may or may not have a sub key, until you get to the lowest key. They hold different types of data to tell the operating system all the information it needs to run properly. This is different from the user files where the working data resides. Windows labels the hard drive as C: by default, and the folder structure is written like: C:\Windows\System32. Almost all Windows components are not case sensitive so the file named System.txt is the same as system.txt.

Linux and Apple have a different type of file structure in that they don't have a registry and use a different naming convention, and although they are slightly different in the folders they contain, they work in the same manner so I will combine them for instructional purposes. The root folder is written with a / and all other folders come next and are written like: /usr/bin. Linux and Apple are case sensitive, so System.txt and system.txt are two different files.

When you are working in the command line in Windows, or terminal in Linux or Apple, you are typing commands and executing them. This is in contrast to using a mouse and opening programs and using icons or buttons. Sometimes it is more efficient to automate repetitive tasks by putting one or more of these commands into a script and executing that, either on a schedule or manually, rather than typing them in one at a time. In Windows, we use Powershell or the command prompt, CMD, for scripting. Powershell scripts have a file extension of .ps1 while CMD scripts are called batch files and end with .bat. Linux and Apple use Bash (Bourne again shell) or one of the equivalents such as Ksh (K shell) or Csh (C shell).

Python is also a scripting language, although it is a higher level language and has more features than a shell and uses libraries and modules as well. HTML (hyper text markup language) and CSS (cascading style sheets) are used to display web pages. There are also others used such as Javascript, XML (extensible markup language), PHP (personal home page), Ruby, and many more. Since this is an introduction to scripting, these are mentioned for familiarization.

One feature of an operating system which you will come to rely on is error reporting. Whenever something happens on a system, a log entry is made. On a Windows machine, these are viewed using the event viewer, which is a database of log entries. The event viewer shows five sets of logs: application, security, setup, system, and forwarded events. Application logs are where the logs of programs go. Security logs are where you will find logs of users logging into the system and whether they failed or succeeded. Setup logs show information and errors from the OS installation. System logs show information and errors which happened to the system and not the programs installed. Forwarded events are where events forwarded by remote systems are stored.

The events have four levels of severity: critical, warning, error, and information. Verbose is also listed, but that is not a level of severity but refers to the amount of information given. For troubleshooting, you only need to look at application and system logs mainly and sometimes security logs. In order to filter out the information logs, I like to right click on one and create a custom view with only critical, warning, and error selected. If you don't find your problem using these, then you can go back and use the time and date of the errors to find informational logs to see what was happening at that time. I only use the verbose setting when I am having no luck with the others.

Linux and Apple use a different method of logging. Instead of a database, they use text files which have the latest log entries written to them and saved in specific locations. In Linux, the logs are stored in /var/log while in Apple there are two locations,

/system/library/SystemDiagnostics/logs and /library/DiagnosticsReports/logs. Since both Linux and Apple are case sensitive, spelling matters in this case. They both store their error logs in plain text files for ease of reading.

When reading error logs, if you can't interpret what something means (even if you can you may not be correct), it is a good idea to copy then entire message and search for that using your favorite search engine. Google is pretty good for this, but I hate them and refuse to use them whenever possible, so I use duck duck go. The main thing to remember about troubleshooting is to always follow the same procedure unless you know what the issue is having dealt with it before. When searching for the error message, if you get into the habit of always keeping a small text file with any url's you found with information on how to fix it and any changes you made to the computer, it will become your own personal knowledge base. We will discuss that further in the troubleshooting section.

Software installation is pretty easy on all modern operating systems. On WIndows, you either use a .exe, .msi, or another installer file which are less common, but the .msi is the recommended type. It has a relational database that is used by the installer. If it has to write data to certain protected directories, it needs administrator permission. When it does this, whatever it saves will be saved under the administrator profile so it may not be where you expect it to be. On Linux, you just use the package manager, either through the GUI (graphic user interface) or the terminal. For Apple it is similar, although for either Apple or Linux you can also just click on the installation files and it should work, but not in all instances.

Compiling software from source code is a little more complex, but is easily accomplished. You have to make sure to have the proper tools installed and then it just depends on having the build environement set up properly. When you have a terminal opened and are in the same directory as the source code, you untar the source, change to the directory it made, run make, then make install. Those are the basics and some software and distros require slightly different commands, build environments, and tools so I am going to leave the details out in order to simplify the process. A big advantage of building from source is security, since you can look at the code before it is installed to make sure there are no back doors or malware being installed. A good way to get familiar with Linux is to use Linux From Scratch and build a working operating system from source code.

Networks are what we use to send and receive information. They can be made up of different types of wire or fiber optics inside cables which make up the physical layer. The most common type of network is made of copper wire; either phone lines, coaxial cable, or ethernet. Ethernet is used inside to carry the data from routers or switches to devices, while cable and phone lines are used to carry the data from one location to another. Fiber optic cable can be used in both instances, although it is usually used for just backbones inside networks and ethernet is used to carry data to and from devices. The data itself is nothing more than pulses of electricity, light, and radio traffic (in wireless networks); on for a 1 and off for a 0. The rest of this just explains how those pulses are moved and interpreted.

Historically there are several different types of network topology, or the way they are arranged to pass traffic, but I will stick to the ones you will be working with for simplicity. You will most likely run into star networks. Some of the others you may see are tree, mesh, and hybrid. Star networks have all the devices connected to a central hub. Tree networks are similar, except the hubs then connect to a control hub. Mesh networks have all the devices connected to one another and are similar to what the internet looks like. Hybrid networks are just a combination of two or more other types.

To move data outside your LAN (local area network) you need a modem to send and receive; a cable modem for coaxial use, a DSL (digital subscriber line)modem for DSL use, a T1 modem for T1 lines (the traffic is passed over phone lines like DSL) or a fiber modem for fiber optic use. Modem stands for modulator/demodulator, changing binary encoded data to electricity or light. When it encodes the data to send out it modulates it and when it receives data it demodulates it.

After the incoming data passes the modem, it goes to a router which then routes the packets to the device they are intended for in your home network or a switch at most commercial and large sites. The router does several things: it assigns each device on the network an IP address, keeps a routing table of devices on the network and places outside the network it communicates with, filters traffic, and transmits and receives all network traffic.

A switch allows you to connect a lot of devices to one port of a router. It can be either unmanaged (dumb) or managed (smart). Unmanaged switches allow the router to send traffic to individual devices. The routing table keeps track of where each device is plugged in. The big advantages of unmanaged switches is they are cheaper and they are plug-and-play. The downside is they have no ability to monitor the network and they lack some advanced features such as VLANs (virtual local area network).

Access points (AP) are how we connect wirelessly to the network. Some home routers offer wireless capabilities and even modems in one unit. An AP has its own IP address and allows a device to connect to a router wirelessly. After connecting, the router is what assigns the IP address to the device, not the AP. The AP is just performing as a repeater, receiving and transmitting information it gets from the devices to the router. Sometimes an AP is configured to serve IP addresses as well.

Antennas are how we send and receive wireless signals. They use the radio frequency of 2.4 and 5 GHz (giga hertz). They are either omni-directional - transmit in all directions at once like the paddle antennas on a home wifi router, or uni-directional - which transmit in one direction like a yagi antenna. There are several protocols used to transmit wifi, listed in order of development: 802.11b, 802.11a, 802.11g, 802.11n (wifi 4), 802.11ac (wifi 5), and 802.11ax (wifi 6).

There are seven layers in the OSI (open systems interconnection) model of a network and they are: 1 - physical, 2 - data, 3 - network, 4 - transport, 5 - session, 6 - presentation, and 7 - application. When we refer to the OSI model, this is how we figure out what is happening to data based on what layer it is passing through. For example, at layer 1, the data is binary and if traveling over ethernet, it is low voltage DC around 2.5V to 5.5V (depending on how it's measured) for a 1 and 0 for a 0. Some switches provide power over ethernet (POE) to provide power to things like access points and security cameras which ranges from 37V to 57V.

The data is organized into packets by protocols or applications at the different levels. These are controlled by standards that tell each protocol how to organize them. For example, an internet protocol called UDP, user datagram protocol, has packets which are put together as follows: it has a header of 8 bytes and then the data which can be up to 65,507 bytes since it also has an IP header of 20 bytes. Each layer from 4 through 7 are on your computer, 2 and 3 are the router and switch, while 1 is the modem and wires, fiber optics, or radio signals. Each protocol has standards defined so that they are able to communicate with one another and other devices.

Say you want to send an email. The email application takes the data at layer 7 and sends it to layer 5 (layer 6 is the presentation layer, where you see it presented to you) where it gets encoded and sent to layer 4. The transport layer is where TCP, transmission control protocol, lives. This is similar to UDP mentioned above, and is how the data is prepared for transmission. After the data is encoded, it has a header with the destination and checksum that is sent to the router. The router adds a header and sends the data along, then the modem takes the 1s and 0s and

converts them to electricity or light and sends them to the other side where the process is reversed and the email is read.

In order for data to move, it needs sent from one place to another. That is what a router does, it routes the data to its destination. There are two types of routing: static and dynamic. Static routing uses pre computed routing tables to figure out where to send data and is usually used in small networks which don't change much, if at all. Larger networks and networks which change often use dynamic routing. Some examples of dynamic routing are OSPF (open shortest path first), RIP routing information protocol), and EIGRP (enhanced interior gateway routing protocol). Some of the ways routers determine which paths to use include distance vector algorithms, link state algorithms, and optimized link state algorithms.

A LAN (local area network) is local to one location, usually a building or single floor of a building while a WAN (wide area network) includes different locations or buildings, different cities, or states. There are routers in datacenters that are responsible for moving traffic over large distances. They are called backbone routers and usually operate using BGP (border gateway protocol) to handle massive amounts of traffic efficiently. They are commonly operated by your ISP (internet service provider) or companies that move traffic exclusively. An SD-WAN is a software defined WAN that can be used in remote locations to connect different locations so the routers think they are in the same location.

DNS. It's always DNS. It stands for domain name system and it's how we use the internet with web browsers. When you type happy.com into the address bar of a web browser, several things happen. One of them is that the name changes from words to numbers. IPv4, or internet protocol version 4 uses four sets of numbers from 0 to 255 and are called octets. Since it is easier for people to remember yahoo.com than 98.137.11.163, we use DNS to convert it.

Think of it like telephone numbers. Instead of remembering the phone number, we remember the name and look it up in a phone book. But before we go to the phone book, we check the fridge to see if it’s there, and if not, in the contacts of your phone. If it’s not there, then we grab the phone book. For DNS, first the computer looks in the local DNS cache, and if it isn’t there, then it goes to the one in the router. If the router doesn’t have the name resolution cached, it goes to the configured DNS server, or the phone book of the internet.

IPv6 or internet protocol version 6 uses 128 bits instead of 32 like IPv4 uses, so instead of four octets, there are eight sets. It uses a colon : to separate them instead of a dot . And where IPv4 was numbers, ipV6 is hexadecimal numbers. The IPv4 block is out of useable address spaces or

almost out, I can’t keep up. It has 2^32 address spaces, which is just over 4.3 billion. IPv6 has 2^128, or a whole lot more than that, 240 trillion followed by 24 zeroes. We are in the process of converting to IPv6, but it has taken a long time.

The cloud is just someone else's computer. Well, a little more to it than that but not much. Just a bunch of servers in a data center, or onsite at the office, or even a mix of the two. This leads us to the names, cloud, hybrid, or on-premesis. Then there are public clouds and private clouds. Security is handled by the data center, but availability is covered under a service level agreement (SLA). Recently there has been a trend toward things as a service (*aaS) like software or security and they are handled by managed service providers (MSP) in operation centers using remote monitoring software.

Virtualization is becoming increasingly popular, given the ease of deployment and configurations. You can clone machines or take snapshots and use them to revert back when needed or as a backup. There are several different types of virtual machines: Virtualbox, VMware, Parallels, and QEMU and most likely others I forgot. This is different than sandboxing, or running programs in an isolated environment from the operating system. They help prevent infections from spreading when used properly. Docker images are a decent way of isolating things as well. They are packages that contain everything needed to run a program. They come in handy because they are self contained and small. They are also easy to make and use.

Troubleshooting is how we fix things. In order to be effective at it, we need to be consistent and use a methodology. In other words, do the same thing and use the same techniques every time. There are exceptions to this, such as when you see a problem you have seen before you may want to try the thing that fixed it last time rather than starting with anything else. As a beginner, it is important to develop a good methodology and stick to it, while documenting your work thoroughly. If you keep good notes on your work, you can save a lot of time by searching there first if you have already solved that problem. Since it is very difficult to remember all the details you need, documentation is vital.

The first thing you should do when talking to the user is listen completely so that you gain an understanding of exactly what the problem is. Do this by having them explain the problem completely and walking you through the steps they took to get there. Sometimes you'll find that they are doing something wrong or that a process has changed. The easiest way to do this is to just say, "Show me what you are trying to do". Then you watch what they are doing and pay attention to what is going on. Sometimes you will be troubleshooting a program that you know nothing about. This is the best way to get a quick overview of what is happening

and what should be happening. Sometimes by explaining the workings of the program, the user realizes their mistakes and fixes the problem themselves. If that happens, write it down, because you may see that again. Look for any error messages that pop up and note what they say exactly. Take a picture or screenshot if possible. Pay attention to the time on the computer so you can verify the errors in the error logs when you look at them.

If you have a clear error message and have noted any actions or other details, you can skip searching the logs right now and simply search on the internet for the error message. Someone somewhere has had that problem before and they may have found a solution and left it on the internet for you. If they didn't or if what you found doesn't work, then it's time to work. Search the error logs, paying attention to which ones to start with. On Windows computers, recall that they are split into system and application logs. For a problem with a specific program, look in the application logs, and when solving a computer issue, look to the system logs. Sometimes you'll find that it isn't clear which ones to start with, such as opening a program and the computer crashes shortly afterwards, but not soon enough that it looks like the program caused it. That could be an issue with an unhandled exception or a system issue that only happens under certain circumstances. If you find one of those, look at each set of logs while correlating the timestamps to see what is happening.

Knowledge base libraries are collections of documents with information on various programs or processes used in the organization. If you have one available, it is a good place to start when troubleshooting something you don't know a lot about. If you are searching the internet, the first thing you'll find is that often having too much information is as bad as or worse than not having enough. At least when you don't find what you're looking for you can still use other methods to solve the problem, such as using logs. When searching, always copy then entire error message including punctuation. When searching for problems without a message, be as clear and specific as you can and be sure to use proper terminology. Include things like model numbers, operating systems, and any programs that are involved. You can also take advantage of google dorks, or search terms that google understands to mean specific things. For example, if you're searching for a .pdf file of a manual to repair Dell computers, you could type: site:dell.com filetype:pdf. You can also use flowcharts or bubble diagrams to learn effective troubleshooting.

Keeping track of what you did to fix something makes you a better tech. Say you have a problem that took you 35 minutes to figure out how to fix. If you keep track of the problem and solution, you can look there at the start of troubleshooting and fix the problem immediately rather than spend 35 minutes looking for something. There are several note taking programs

that index your notes for easy searching. In order to use notes well you need to make sure you keep the correct information and have enough detail to solve the problem but don't keep things that are irrelevant. Posting a link to any forums or webpages you used to find the solution is also a good idea. When I worked at a call center, I kept a page for each day. The program they used is no longer supported, but the ones out there now do a great job of organizing them.

Some programs use a folder structure while others use a database structure. Since most of them hide the abstraction behind the user interface it really doesn't matter what the underlying structure is though. If you take a few seconds to add a keyword system to your notes as you are taking them it will make it much easier to search them in the future. Also, knowing how much detail is enough without putting too much in can take some practice and while learning how to dial it in you will lose some information that just didn't have enough to use. You can fix those by adding the missing information as you find it. If you want more formality you can create your own knowledge base library. There are programs that help with that also, and you could do it manually but may lose some of the functionality unless you are willing to put in some time programming. I copied the entire knowledge base of the one of the companies I worked for and got permission to use them in mine after redacting them. That is on my to-do list.

Collaborative learning for youths. During group learning, mandate a 1.5 minute time limit for speaking, then a 30 second time for deep introspection where everyone thinks about what was said. Then the next person speaks in the same manner. This allows time to engage diffuse and allow the unconscious to work on it while also forcing everyone to pay attention to the words rather than waiting their turn to speak.

When doing group writing projects, have each member read the material, distribute their individual answers among themselves, and come to an agreement on a final finished draft to be submitted with all the individual answers in original format.

Customer service. I can sum up good customer service in two words: be nice. I can also tell you how hard it is to get good customer service from folks, on both sides of the engagement which is why I am devoting five weeks to it. It's not that the customer is always right; most of the time not only are they not right, they often have no clue what they are doing anyway. Not all users are technically illiterate but enough are that it presents a challenge sometimes. And just being technically literate is a far cry from being a tech. This really comes into play when trying to resolve a network issue and you can't connect remotely, also known as 'working blind'. This can get extremely frustrating and you have

to remain professional at all costs. The more familiar you are with the underlying framework, the better you will be at guiding someone to use it.

When you're on the phone, your tome of voice matters. You need to be aware of the inflections and differences between short, terse answers and ones with some explanations. This puts the user in a positive frame by giving the impression you perceive them as equals. They don't need to understand what you're saying, and often will ask questions, so make sure you're well versed in it yourself. This also builds a rapport with the user which goes a long way, as most jobs have customer feedback incentives and ways to measure user engagement. I worked at a company which used a Dalbar system and monitored calls randomly to score them based on how well they did on customer service. I got a perfect score on a call once, something hardly ever done there.

One of the things we had to do was engage in conversation while working and waiting on reboots or things to load or scans to run. Mostly explaining what you were doing and why, but also general chit-chat while avoiding controversial things like politics. Pets and family were always a decent choice, or hobbies and sports. Another thing that helps customer service is thoroughness. You don't want to get into a habit of not resolving an issue completely because it would take too long or telling them to reboot and call back when it was up again, then marking it resolved. Two of the metrics companies use to rate your work are call time and first call resolution.

Seriously, as I am writing this lesson plan I keep thinking about the book called 'All I really need to know I learned in kindergarten'. Things like: use your manners, be polite, try to be as helpful as you can, etc. Normally at this point, I would advise you to just say things like "yes sir" or "no ma'am", but today there are preferred pronouns. This can go two ways: you ask and while satisifying those who prefer a different one you are offending one who has strong preferences against them or you don't ask and reverse the offense. I would err on the side of using your best judgment and if you have information pop up when you get a call, go with that. If the name is Samual and it sounds like a man or even woman, I would use sir and apologize if I were wrong. I think what it comes down to is using the respect and even if you're wrong it was done with respect. Just take care of it and move on with the call professionally.

One thing you will encounter is the user who is mad. How you handle that depends greatly on the circumstances but there are some things to generally keep in mind. Try to calm the user, this varies depending on the reasons given and how mad they are. If you feel you are being verbally abused, tell the user that they are being abusive to you and need to calm down or you will end the call and do so if necessary. Then tell your supervisor to pull the call so they can address it then. If they

calm down, try to keep them calm by discussing something unrelated when you get a chance, but don't take away time from troubleshooting for it. It is vital in these cases to appear as if you were the best in the world at this, even if you are new. Do this by talking and explaining what you are doing.

One thing you want to avoid is using acronyms and use proper terminology. Even if they don't know what you are talking about when you clear the print spooler, they know the last tech did that when it happened and wonder why you are doing something different. Always use proper language and this includes respectfully dealing with everyone as well as correct grammar, perhaps even more importantly. People will forget what you say but they will never forget how you made them feel. If you end up making the user feel like an idiot because they had a simple problem; that is all they will remember about you, not that they forgot to hold shift down one time.