Teaching Computer Science in 11th Grade

Charles C. Weems

The cosmic forces thus utilized will lead to strange machines, but ones that will reduce human labor, since they will possess a certain intelligence. Cosmically oriented spiritual science will have the task of reducing harmful effects caused by the great temptations that arise from those mechanized beasts, which the people themselves have created. And we must add something else to all this: it is necessary for people to prepare by seeing reality for what it is, not as illusions—by truly viewing the world in a spiritual way.

– Rudolf Steiner, *Spirit Beings and the Ground of the World*, 3, Dornach, November 25, 1917

There is often a dramatic change in students between 10th and 11th grade with respect to their capacity to work with abstraction. In 9th grade, we consider computers concretely, working toward an understanding of what a computer is, and how it is fundamentally different from the human being. In 10th grade, we more actively engage the concrete aspects by dissecting computers, seeing how they are built, how they work, and how they are programmed at the lowest level. The content for teaching these courses can easily be learned by anyone with a background in math, logic, or the "maker" hobby.

In 11th grade, students start to look forward and outward into the world, and much of their world is online. They are denizens of the internet, and feel at home in it, even though they don't fully understand it. Its virtual alternative to reality presents them with many temptations, mostly of an illusory nature, and in this 11th grade computer science course, we begin preparing them to distinguish it from reality.

Rather than preaching about dangers on the internet, we can explain how it works and why. In the process, they can discover the means and motivations of people who use it to exploit others and the necessity of developing a new sense of morality to guide them. To teach this content, the teacher must delve more deeply into an understanding of the technology that stands behind the modern online world. This article, thus, begins with an overview of the relevant concepts.

One of the student's most common experiences is performing a web search. But how could Google possibly search the entire internet in an instant to return links to hundreds of thousands of silly cat videos?

The answer is that it doesn't. A search engine gets its data from a program called a web crawler that continuously follows links between web sites, returning the pages to another program, called an indexer, that summarizes the data as a set of numerical values, somewhat like the call numbers in a library catalog.

Rather than preaching about dangers on the internet, we can explain how it works and why.

Old copies of a page are kept (or "cached"), so that when a page is returned by the crawler, the indexer can first look to see if anything has changed. The crawler runs on a vast number of computers at once, reindexing the web many times per day. That is why, when you put something on the internet and then delete it after having second thoughts, it may already have been indexed and cached, so it takes on an independent existence beyond your control.

Because the web is already indexed, when we enter a search query, it is just a matter of generating index values from the words in the query and matching them to the recorded indexes. Each index value connects to the previously established list of pages that match, so it takes a fraction of a second to return the list. But the list we see isn't random. Pages in the list are ranked to give us the most relevant results first.

Students can identify many of the factors that can go into a page ranking: the number of other pages that link to a given page; the number of keywords on the page that match words in the search; how many times other users have gone to the page when it has been listed in search results (called click-throughs); the location of the searcher, etc.

In considering these, it becomes clear that manipulation of page ranking can be used to subtly alter a user's perception of the world. Google Bombs are obvious examples that illustrate exploitation of ranking. These are search terms that return unexpected results because someone has tricked the ranking algorithm into registering certain pages as more popular than others. For example, during a previous presidential election, the search "completely wrong" returned a large number of images of one of the candidates.

Ranking can also be used to favor certain online retailers, including those who pay for placement. A useful exercise is to do the same search using different search engines, such as Bing, DuckDuckGo, and Yandex. Many of their results will be similar, but some will be different. In some cases, different students will see different results from the same engine, which can be due to search history, geographical location of the computer on which the search is done, or the language the user usually uses when working with this computer.

Why are search results individualized? In part, to maximize effectiveness of advertising. When a search company can claim a higher rate of click-throughs, they can sell more ad space. More relevant results also keep users coming back. But they also lead to users seeing information that is compatible with their existing world view, which tends to reinforce their beliefs.

The result can be what some psychologists call "cognitive bubbles." A person is tempted to believe that a one-sided perspective is reality because their searches avoid sites with alternative views. Some of the current polarization in politics has been attributed to this effect, and it can be seen as dividing people against each other into more narrowly defined identity groups. Students should see that they must develop skills to distinguish reality as it is from the illusory reality that is served up to them by online services.

When you put something on the internet and then delete it after having second thoughts, it may already have been indexed and cached, so it takes on an independent existence beyond your control.

The experience of seeing different rankings shows why it can be useful to go many pages deep into search results to find more diverse sources. This is also a good place to introduce advanced search methods and specialized searches, such as Google Scholar or Patents.

Students can be asked to combine more effective searching with their knowledge of social media, to discover as much information as possible about someone they know and make notes to report in class. A list on the board of the different types of information they find will grow to a considerable length. What is available can be quite surprising. The question of whether they want a college admissions office to have access to similar information about them provides some perspective. Searching for information on themselves, some students will discover that they already have a public presence that will follow them throughout life. Their online activity is creating a kind of "digital Akashic record."

A discussion of how consumers expect online services to be free, even when they are based on billions of dollars of investment, illuminates the challenges of creating a new web-based service, and why a common model for new companies is to develop and prove a service and then sell out to one of the established service providers. In many cases, much of the smaller companies' value is the data it has gathered on its users. Thus, when you provide information online, you don't know who may eventually end up with it, or how they will use or protect it.

Students may want to know how to guard their information online. Some may believe that careful use of privacy settings on social media protects them. However, some social media companies sell user information. Others have complex privacy settings that can be difficult to get right. And law enforcement can request personal information. Thus, some useful advice is not to post anything online that shouldn't appear on the front page of a newspaper.

A common belief is that "I don't have anything of value, and nobody would want to know about me, so I don't have to worry," otherwise known as "security through obscurity." But we never know where life will take us. An example is a congressional candidate whose opponent found a video of him doing a drunken dance in college and turned it into a TV campaign commercial.

To understand the basis for internet exploits, it helps to examine how the internet actually works. We can take students on a tour of the school's network equipment, explaining the various kinds of connections and devices that are encountered. Based on what they observe, students can map the school's internal network.

Even for a very simple network, it will be clear that data from many users are combined before being sent to the internet, because at least one of the devices will have multiple connections coming into it from around the school, and one connection that goes out toward the street. The external connection goes to the school's internet service provider (ISP), which has connections coming to it from many customers in the same area. The ISP sends data over major connections that link to other cities and even other countries. Eventually, a message will find its way down through a similar network on the other end to a specific recipient.

One can then ask students for their ideas about how they would make such a system work. They quickly realize that each end point needs an address, and each message needs to specify a destination address and give its return address.[1] Sending messages back and forth is thus analogous to sending letters through the postal system.

Each message has a section containing the addresses (like the envelope that surrounds a letter) and a section containing the content.[2] Anyone can read the content at any point along the way, so it should be encrypted (with a secret code) to keep it private. And because messages contain return addresses, they can be tracked back to their origin.

For physical, wired, end points, it's easy to envision how addresses could be assigned from a central authority (just as town hall assigns street addresses). But what about mobile devices? In addition to the permanent addresses, we need changeable addresses. A mobile device will request a temporary address from a wireless base station,[3] which has a limited number that it can hand out at any given time.[4] The base station can thus run out of addresses if too many devices request them, which is why it can be difficult to get a connection in a coffee shop where many people are online. The base station recycles the address when the device moves on.

In our mail analogy, you can think of these addresses like the room numbers at a hotel. When you're staying at a hotel, you can have someone send you a package using your room number. But after you check out, someone else can check in and use the same room number.

With wireless in a coffee shop, where the owners may not be careful about the setup, this arrangement can allow a hacker to set up

a wireless station with a stronger signal that overrides the shop's station, and then hand out fake addresses so that messages can be intercepted by his or her computer, with the goal to detect valuable information, such as credit card numbers or user names and passwords.

The internet has many interconnected devices—computers, phones, remote controlled light bulbs—and information must be routed between them according to their addresses. Switches are specialized computers that receive a message from one connection, and send it out on another connection that will get it closer to its destination. More advanced models can be reprogrammed remotely. Thus, there is the potential for hackers to break into switches and install "sniffer" software, enabling them to spy on messages as they pass through.

Students should see from this and the coffeeshop example that it is important to know how to determine when a website uses an encrypted connection, how to set email to use encryption, and, if they have a wireless base station at home, how to ensure that it is secure. It is worthwhile to have a discussion of how the internet is not heavily policed, due to the vast amount of data it carries and comparative lack of law enforcement resources for investigating online crime; thus uninformed users are often unprotected from the many unscrupulous people who invest considerable effort into taking advantage of them.

In addition to the address of a device, a message has to specify the application or service on the device that should receive it. For example, an email application sends a request to download mail from an email service, which sends the response back to the email application on the user's computer, rather than the web browser. Each message thus specifies both the destination address (the device) and the application or service to receive it.[5]

Because computers most easily work with numbers, an internet address is represented by a number.[6] But we don't access sites by number. We use a domain name, such as apple.com. The trailing letters of a name (.com, .edu, .de, etc.) specify its domain, that is, one of the established major types of internet address. When we enter a web site name in a browser, it is translated to a number using the internet equivalent of a phone directory.[7]

First the name is sent to one of a dozen organizations that manage all the names in the directory. That organization sends the browser to another that handles the names for a given domain. It may redirect the lookup to yet another that finally responds with the specific number sought.

Your computer also keeps a list, called a cache, of common and recently accessed addresses. The master directory is updated every fifteen minutes or so, as web addresses are added or changed, but your computer doesn't update its cache as often. When a computer or phone has been offline for a while, there may be a delay while it downloads an updated directory.

The directory system poses a security risk in that a hacked name lookup server can be set to translate a name to a different and possibly malicious destination. For example, in connecting to your bank, a hacked server sends you to a site that looks like your bank, but it steals your username and password. While rare for the big, public servers, a company that runs its own name lookup service internally may not have the same level of security. A more common scenario is that a link in an email or on a web page can look like a normal name (www.mybank.com) but the underlying address goes somewhere else. Students should learn how to inspect a link's actual destination before they click on it.

Students should also have a basic understanding of encryption, which relies on the idea that some mathematical functions are easy to compute but harder to undo. For example, we can multiply a chain of numbers (one of which represents a character in our message) quite easily to get a long number, but given a long number, it's more work to identify the numbers

that were multiplied to get it. If we have given the recipient the numbers in advance (a key to the encryption), then it is easy to divide them out and get our character as the remainder.

Encryption thus relies merely on the assumption that discovering the key is so time-consuming that nobody will invest the effort to do so for our messages. However, it's not impossible—there are government agencies with gigantic computers that can do the job in a reasonable time for select messages.

There is still the problem of sending the key to the recipient without anyone else seeing it. Two students can be given a lock box and a pair of padlocks, each with two keys. The goal is for one student to safely get their lock's key to the other. A little thought and experimentation results in the idea that one student puts their key in the box and locks it, then gives the box to the other student, who adds their lock to the box and returns it. The first student removes their own lock, and returns the locked box to the second student, who can unlock his or her own lock and open the box to get the key.

Our exploration of how the internet works has been leading up to this basic truth: Digital data is divorced from reality. It can be manipulated to represent anything or anyone.

Key exchange is the basis for secure communication on the internet. But what if there is another student, with another pair of locks, between the first two? He or she can pretend to be the counterpart to each of them, and thus get both of their keys and open the box to read its contents before passing it on. This scenario is known as a "man in the middle attack."

How can you be sure the person you are exchanging keys with is not an imposter in the middle? You need a third party you trust. You give that person a key, and they provide a message back to you that says, "I confirm that it really is you who sent me this key," and which includes your key. But the message is encrypted so that nobody but the trusted person can read it.

When you contact the other person, as part of exchanging keys, you send them this message, and they send it to the trusted third party, who decrypts it and compares the key it contains with the key you originally sent. If someone is trying to get between the two of you, using a fake confirmation message with their own key, it won't match the one that the third party has on file for you, and the third party can tell you this.

The internet has about 40 highly trusted organizations[8] that are supposed to manage all of this. But that's too few organizations to handle all the key exchanges on the web. So their job is to certify a larger number of organizations, such as major companies, that can handle the load. And they may in turn certify additional groups. In practice, we tend to rely on third or lower level organizations.

As complicated as that sounds, it comes down to establishing a chain of trust. Students should see that we need organizations we can trust to verify the identities of people and companies we connect with because we can't personally know and develop trust with them ourselves.

But lower level organizations, for which certifying users is not a primary business, may not be as conscientious about checking identities. There have also been cases in which an upper level organization has been tricked through an extensive effort into certifying someone erroneously, as part of an espionage operation. In one case, the goal was to get computers to install an update that appeared to be from Microsoft, but the software[9] actually allowed agents to secretly steal information from the computers on which it was installed.

Our exploration of how the internet works has been leading up to this basic truth: Digital data is divorced from reality. It can be manipulated to represent anything or anyone. A web page can be an honest representation of a business, but it can also be that a page comes up at the top of a search due to a manipulated search ranking and redirects the user to a site with a forged identity

certification. The site may present the illusion of legitimacy and thereby tempts us to give up critical information.

There is no digital solution to the security problem. Encryption is not invulnerable, just costly to break. Identity certificates can be obtained by imposters. Passwords can be discovered with enough effort.

Security on the internet ultimately depends on people being careful.[10] What happens if people fail to do their part? If identity thieves steal enough information about a person, they can get credit cards in that person's name and run up huge bills, file a false tax returns to steal refunds, access bank accounts and drain them, obtain government IDs in their names to use in committing crimes, and otherwise make life miserable for the persons affected.

There is no digital solution to the security problem. Security on the internet ultimately depends on people being careful.

But more often, hackers seek to take control of a computer. That may be a means of obtaining identity information, including stored passwords, or contact lists that can be exploited. But often it is to make use of the computer itself—to store illegal data that the hacker doesn't want to be caught with or to encrypt the data on the computer and then demand a ransom from the user to regain access.[11]

A hacker may also install software that uses the computer to do his or her bidding. That could be sending spam email. Or it could become part of a coordinated army of compromised computers for carrying out online attacks on command.[12] Malicious software, called malware, comes in many forms, each with its own way of taking over a computer, which students should know how to defend against.[13]

One of the weakest points in computer security is the user. The "hack" of the Democratic National Committee during the 2016 presidential campaign and the theft of email, was the result of one campaign official's believing an email that told him his password needed to be changed and he clicked on a link that went to a fake password reset page where he was asked to enter his existing password. The hacker immediately logged in with that password and took control of the machine. This unsophisticated but effective approach is generally known as "phishing." When carefully directed at a specific person, it is called "spear phishing."

Even without all of the technical trickery, sites can tempt us with a range of offerings. Free videos, music, games, apps, and illicit content are common lures. Students should consider how these sites support themselves. Some may do so through advertising, but some collect information that can be sold or infect computers with malware.

One of the best defenses on the internet is to be skeptical of anything that appears to be free, or any message that tries to instill fear so we will take some action. Students should know that banks, credit card companies, the IRS, etc., will never send a link to a login page.

Some people think that if they create a false identity (mail account, social network account, etc.) with a different name, address, and so on, then they can use the internet anonymously to access illicit content, post controversial remarks on social and other media, engage in bullying, lure young people into harmful activities, etc. The possibility of anonymity can be empowering, but it can also bring out the worst in people.

However, it is very difficult to be truly anonymous on the internet. Return addresses in messages make it possible to trace their sources. The web is constantly monitored by software that has purposes ranging from improving the flow of messages to identifying shopping trends. Many sites, including search engines, track the places users are accessing to determine their interests and sell the data to advertisers.[14]

Google once ran a Super Bowl ad which showed a series of searches revealing a story of someone going on exchange to Paris, falling

in love, getting married, and having children. While most people thought it was cute, internet privacy experts saw it as chilling. Eric Schmidt, as chairman of Google, once said, "We know where you are and what you are doing, and we have a pretty good idea of what you are thinking."

On the surface, the purpose of this portion of the class can be seen as educating students about why they must be careful on the internet, and how to avoid common traps. But one can also have a discussion of how technology calls us to rise to a higher moral standard while it gives us the potential for varying degrees of anonymity and power, tempts us in many ways to do things we might not otherwise consider, makes it possible for others to reveal our actions long after we may have forgotten them, and exposes us to a wide range of risks.

In the modern world, we cannot escape using the internet. We must develop a new level of consciousness around our actions so that our choices are thoughtfully guided from a moral basis. If we accept this challenge, then technology and the internet can actually be seen as helping us take the next step in the evolution of human consciousness.

The class can shift to a new focus: Why is it so hard to get software to work correctly? The first thing to discover is the challenge of expressing the solution to a problem algorithmically. For example, the teacher can pretend to be a robot that the students must direct to do some task, such as going to a fountain to get a drink of water. As the teacher gets stuck in corners, knocks furniture over, or otherwise turns his or her instructions into literal actions that were not anticipated, students learn the necessity of planning and expressing a sequence of steps carefully. They will also discover the limited vocabulary the robot responds to and devise ways to express repetition and possibly decision making.[15] Students can discuss the experience afterward and be invited to describe the change in mindset that it involved.

If there are computers available, the students can be given a simple program to implement. For example, they can start with the traditional 'Hello World' program, which pops up a message that greets them.[16] Then they can extend the program to input some text that becomes part of the message.[17] Once they have this working, they have most of the mechanics necessary to create a MadLib program.[18] They can choose a story from a collection of Mad Lib books and build the program by replicating pieces of the earlier program. It takes very little knowledge of programming to succeed because it is nearly all a matter of repeating a simple pattern.

It's not about learning a programming language… [I]t is having the experience of how much effort and attention to detail it takes to make software flawless.

The Mad Lib stories that are displayed can be exceptionally funny, and the students will take delight in trying each other's creations. Along the way, however, they will discover that a programming language is extremely strict and will refuse to compile a program if a single quotation mark or semicolon is out of place. Even after it runs, they often find that the spacing in their output is messy and that their use of line breaks is ragged. It takes quite a bit of effort to get everything just right, because the MadLib program, while conceptually simple, is long enough to have multiple errors.

That is, of course, the whole point. It's not about learning a programming language, although it helps their comfort level to explain what each line of code does. Instead, it is having the experience of how much effort and attention to detail it takes to make software flawless. Once they appreciate this, it can be pointed out that the security of the internet depends on hundreds of millions of lines of code all working perfectly. Airliners, hospital equipment, the electric power grid, nuclear reactors, and many other life-critical systems depend on this same level of perfection.

Hackers, on their end, will intensively examine code, looking for errors they can exploit.

The other aspect of the experience that the students encounter is that time seems to vanish while working on a program. After spending an entire period repeatedly compiling, fixing, and testing their code, when the bell goes off, many will express shock that the class is over.

Some will beg to stay and finish, revealing how programming can become addictive. The computer draws us into its world, made of simplified rules, which it seems we can completely master, unlike the real world. It gives us a feeling that here we have a powerful, yet slave-like machine into which we can place our intelligence, so that it becomes an active representation of our selves.

With some reflection, students should realize that creating a program can be a very egotistical experience. Yet, the best software is developed entirely with the user in mind. When software is hard to use, or contains obvious flaws, it is a sign of a programmer's ego overriding good judgment. Again, the technology calls us to rise to a new level, burning out our egotism, so that we are motivated by compassion for the people who will make use of our efforts.

Why do we have an internet? The purported purpose is communication, but to commune with another is to share thoughts and feelings even to a spiritual level. The idea that we can truly communicate using digital data is thus a materialistic illusion. But if we see the internet as a development that offers humanity regular opportunities to exercise a higher form of judgment and volition, then we grasp a spiritual reality that calls upon teachers to guide students in avoiding the temptations that arise from this beast and to respond to it in ways that lead them upward on the evolutionary path of humankind.

In twelfth grade, we will more deeply explore who we are with respect to our technology, as individuals and as a society, and then contrast human intelligence with artificial intelligence.

ENDNOTES

1 At this point, it's worth noting that the internet has many such conventions, called protocols. The one the students just reinvented is called the internet Protocol (IP), and the addresses are called IP addresses.

2 A message actually has three sections called the header, payload, and checksum. The checksum helps detect errors in transmission, much the way that a zip code is a double check of the city and state information on a envelope. The header contains not just the addresses, but also other control information, such as the length of the message and time it was sent. Payload is another name for the message content.

3 Also called a wireless router.

4 Permanent addresses are said to be "static" and changeable addresses are "dynamic." The Dynamic Host Configuration Protocol (DHCP) is responsible for handing out temporary IP addresses. When a mobile device requests an IP address from a base station, it uses DHCP to allocate one from a set it owns. When a device stops using the address, it gets recycled.

DHCP presents a potential security risk because a device with a stronger signal than the base station can pretend to be the base station and hand out addresses that cause messages to go through the device, allowing it to spy on them. Running the "traceroute" (Unix, MacOS) or "tracert" (Windows) network utility for a web address shows that traffic goes through many intermediate computers and switches.

5 Applications and services are assigned "port" numbers according to the terminal connect protocol (TCP). For example, email might use port 993, while the time service uses port 123, and a browser uses port 80. If IP addresses are analogous to street addresses, the TCP port is akin to an apartment number. The combination of the IP address and the TCP port number serve to uniquely identify the destination for a message. Thus, we often see references to the internet using TCP/IP. Most people do not use the majority of port numbers, since they only use a few applications. Even so, the convention is that the messages still need to be delivered, which could enable hackers to deposit malicious code into a computer's memory. Thus, operating systems allow users to selectively disable all of the numbers that are unused, setting up what is known as a firewall. Students should see how to check their computer's firewall settings.

6 The number is usually written in four parts. For example 17.178.96.59 is an address for Apple Inc. More recently, as the world has started to run out of numbers, we have started moving to IP version 6 (IPv6), which uses longer numbers, written in base 16. For example, 030a:74d5:f12c:1003:299b:8080:feed:0042.

7 The directory is called the Domain Name System (DNS). The internet has several hundred top-level DNS servers that are controlled by about a dozen different organizations that manage the registration of names. Each name is part of a domain (e.g., .com, .org, .edu. .uk, etc.)

8 These top level organizations are known as root certificate authorities; all they do is certify other major organizations as subsidiary certificate authorities. It costs quite a bit to become certified, so some organizations, such as smaller companies and schools, may declare themselves to be self-certified, and then they can issue certificates to computers within the organization to enable secure communications between them.

9 Malware called "flame."

10 That means certificate authorities have to carefully check identities. Programmers have to write secure, bug-free code. Web sites must ensure they are using software that hasn't been compromised. Users need to manage their security settings; watch carefully to ensure they are using encrypted communication with sites that have valid certificates and that links are legitimate before clicking on them; watch for unusual behavior on their computers that may be due to a virus infection; and much more. When students grasp the number of points of potential failure, and compare that to the effort that hackers are willing to invest in finding weaknesses, they will see why it is nearly impossible to avoid having data stolen. Thus, it is important to consider carefully the information they put online.

When students grasp the number of points of potential failure, and compare that to the effort that hackers are willing to invest in finding weaknesses, they will see why it is nearly impossible to avoid having data stolen. Thus, it is important to consider carefully the information they put online.

11 This is called encryption extortion, or ransomware.

12 The army is called a botnet, and it can be used to attack and shut down sites (perhaps for political reasons) by flooding them with requests called a "distributed denial of service," or DDoS, attack.

13 More specifically, a virus is a piece of software that is run when files are shared. Like a real virus, it is spread by contact between infected machines. A Trojan horse is a program that seems to do one thing, such as giving the user a free computer game, but that infects the computer with malware when installed. A rootkit is malware that takes over the operating system at the deepest level, so it is impossible to remove without completely wiping the machine. A rootkit might include a keylogger, which records everything typed on the computer, sending passwords and credit card information to the hackers. A worm is malware that actively scans the internet for other computers that have security holes, such as open TCP ports, and then copies itself onto them, turning them into "zombies" that continue to spread the worm. Unlike a virus or a Trojan horse, a worm can infect a computer without requiring the user to do anything. Zero day exploits are security holes that the computer makers haven't found and fixed. A nightmare scenario for computer security experts is a zero day exploit that enables rootkitting, and can be spread by a worm, since it could silently and rapidly infect millions of computers. In May of 2017, over 200,000 computers were infected in this manner with ransomware called WannaCry.

14 Browsers maintain caches of recently downloaded site content, history of accesses, and collections of files called "cookies" that enable sites to identify return visitors. Many browsers automatically share this information with their user's other devices, so that, for example, an access from a phone can be easily resumed from a laptop, and a copy of all of it is held on a "cloud" system. Servers keep logs of all access attempts to help identify the sources of attacks. Even in "private browsing" mode, most computers expose enough information (make, model, operating system version, browser version, location) to have what is effectively a digital fingerprint that enables them to be tracked. There are tools that do enable people to be relatively anonymous, but they take considerable knowledge and effort to employ effectively. They are agnostic as to whether they are enabling criminal activity or free speech. If someone is investigating their use, it should be a trigger for reconsidering the motivation behind wanting to be an anonymous actor on the internet.

15 The basic organizing mechanisms for the instructions in an algorithm are sequence, branch, and loop. Because computers on the internet do different things at the same time, there is also a notion of instructions working in parallel, which has become an important aspect of programming. Coordinating parallel programs can be very challenging, as can be demonstrated by having two students work side by side at the board to quickly add one to numbers in a list. Very soon, it will be seen that they can't keep track of which ones have already been incremented, and so some will have two added by mistake. Learning to cope with this kind of interaction helps preserve a flexibility of thinking that can be lost if students are only taught to turn problem solutions into sequences of steps.

16 Here is a Java "Hello world" program:

```
import javax.swing.*
public class HelloWorld
{
  public static void main (String [] args)
  {
    JOptionPane.showMessageDialog(null,
   "Hello world");
  }
}
```

The BlueJ environment (www.bluej.org) is a free, student-oriented, Java development platform that works identically on Windows, MacOS, and Linux.

17 Extended program that inputs text to replace "world":

```
import javax.swing.*
public class HelloName
{
  public static void main (String [] args)
  {
    String name = JOptionPane.
showInputDialog("Enter a name: ");
    JOptionPane.showMessageDialog(null,
"Hello " + name);
    }
  }
```

18 An example of a very short MadLib-like program:

```
import javax.swing.*
public class MadLib
{
  public static void main (String [] args)
  {
    String adjective1 = JOptionPane.
   showInputDialog("Enter an adjective: ");
    String noun1 = JOptionPane.
   showInputDialog("Enter a noun: ");
    String verb1 = JOptionPane.
   showInputDialog("Enter a verb: ");
    String adjective2 = JOptionPane.
   showInputDialog("Enter an adjective: ");
    String noun2 = JOptionPane.
   showInputDialog("Enter a noun: ");
    JOptionPane.showMessageDialog(null,
   "The " + adjective1 + " " + noun1 +
      " " + verb1 + " over the \n" +
   adjective2 + " " + noun2 +
      " and said, \"Oh my!\"");
  }
}
```

This program will ask the user to enter an adjective, a noun, a verb, an adjective, and a noun. If the user enters: "blue," "cow," "flew," "fat," and "donkey," it will output:

```
The blue cow flew over the fat donkey
and said, "Oh my!"
```

Charles C. (Chip) Weems, *PhD, is a professor of Computer Science at the University of Massachusetts, Amherst, and teaches computer science, geology, meteorology, and astronomy at Hartsbrook High School, in Hadley MA. He is a member of the Pedagogical Section of the Anthroposophical Society, a co-author of 28 introductory computer science textbooks, and has led workshops at national AWSNA conferences on technology as it relates to the Waldorf curriculum*.