ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #16679

Introduction to Computing Technology: New Interactive Animated Web-Based Learning Content

Prof. Frank Vahid, University of California - Riverside

Frank Vahid is a Professor of Computer Science and Engineering at the Univ. of California, Riverside. His research interests include embedded systems design, and engineering education. He is a co-founder of zyBooks.com.

Dr. Susan Lysecky, zyBooks, Inc.
Dr. Alex Daniel Edgcomb, University of California - Riverside

Alex Edgcomb finished his PhD in computer science at UC Riverside in 2014. Alex has continued working as a research specialist at UC Riverside with his PhD advisor, studying the efficacy of web-native content for STEM education. Alex also works with zyBooks, a startup that develops interactive, web-native textbooks in STEM.

Introduction to Computing Technology: New Interactive Animated Web-Based Learning Content

Frank Vahid, Computer Science & Engin., Univ. of California, Riverside (also with zyBooks)
Susan Lysecky, zyBooks, Los Gatos, California
Alex Edgcomb, zyBooks, Los Gatos, California (also with UC Riverside)

Abstract

Computing technology is ubiquitous, in the form of computers, tablets, smartphones, the web, cloud computing, email, text messages, social media, and much more. Computing technology is centrally relevant in the careers and lives of increasingly more people. As a result, more universities require students to take at least one course in computing/information technology. Unfortunately, the most widely used textbooks are written as (or have bloated into) comprehensive references, focusing excessively on terms and definitions that are quickly forgotten by students after a course is over. Instead, given computing technology's central relevance, such a course should be one of the most interesting and useful at a university, being an engaging introduction that positively influences students' interactions with computing technologies. Thus, in 2015, we developed new learning material for computing technology courses, starting from scratch, and developing natively for the web. We favored meaningful understanding of the most relevant computing technologies, versus exhaustive but shallow coverage of a wider topic array. This paper highlights the first version of the learning material, including discussions of some of the key interactive web-based activities. In its first year, the material has been adopted by over 20 universities, either for a full course, or as part of a course introducing computer science or computer programming (the material is modular and thus can be combined with other materials), and feedback from instructors and students is very positive. The material is currently published as "Computing Technology for All" (zyBooks.com).

Introduction

While a computing technology course should probably be one of the most interesting and useful courses on campus, the opposite is often true. The most widely used textbooks from major publishers focus on serving as comprehensive references rather than interesting introductions, with extensive lists of terms and definitions, such as B2B (business-to-business) versus B2C (business-to-consumer) e-commerce, or ROM (read-only memory) versus RAM (random-access memory). As such, lectures tend to focus on terms and definitions, and quizzes and exams emphasize memorization of such items. The result is a dry course, with little retention of concepts afterwards, little improvement in a student's insight, and little change in a student's interaction with computing technology.

Thus, in 2015 we invested extensive effort on creating new interactive web-based content for introducing computing technology. We focus on having students learn the fundamental concepts in computing technology, and avoid attempting to simultaneously be a comprehensive reference. Each concept is taught interactively. Animations are used to visually and dynamically convey key ideas, such as an Internet packet moving across the web, a disk drive spinning to retrieve data that is then stored in local RAM, or a cloud server experiencing a denial-of-service attack. Hundreds of learning questions are embedded throughout the content. Various tools appear within the content as well, such as an HTML editor that immediately renders a preview. Content is supplemented by powerful and popular online tools, such as a download speed testing site, as well as by current news articles on privacy and other subjects. Furthermore, for courses that wish to introduce some programming, the material includes a Python programming window directly in the Python sections, bypassing the technical hurdles that often occur if students must install their own Python environment.

This paper describes the various topics covered by the interactive web-based content for introducing computing technology (history, hardware, software, the Internet, the web, operating systems, computer applications, web applications, mobile apps, privacy, security, and more), and illustrates the various web-based interactive elements and tools. The material is published as "Computing Technology for All" (zyBooks.com). First released in summer 2015, the material has been used by several thousand students at over 20 universities.

Topic overview

The first version of the material covers the core topics found in most textbooks on computing technology. Chapters include:

- History and Basics
    - A brief history of computers, what computer programs are, the ubiquity of computers (servers, laptops, smartphones, embedded systems, etc.), how information is represented as bits, and computing's relevance to many careers. The chapter strives to have students get a real sense of how recent the advent of computers really are, how rapidly computing/information technology has grown, and how the future of computing is surely to continue growing.
- Hardware and Software
    - Basic computing hardware, cache/memory/drives, computer types (servers, desktops, embedded, etc.), input/output devices, and Moore's Law. Extensive animations, and tools like a built-in "turtle" drawing tool for teaching programming, strive to help students gain an intuitive understanding of what a computer does and how programs work.

    - Uniquely, the material introduces machine language and assembly language, in a simple and understandable way for novices using a highly-simplified 5-item instruction set, including built-in hands-on easy-to-use tools so students see how each instruction executes. The material introduces high-level languages also, and includes a built-in hands-on brief intro to Python as well, where students can immediately run an existing program and modify the program to see impacts.
- The Internet and Web
    - Internet basics, IP addresses, home networking, web basics, search engines (and tips), domain names and URLs, and setting up a website. The material uses extensive animations to show, for example how packets travel from a user's device to servers and back, or how a domain name differs from and relates to a hosted website. The material includes concrete steps for registering domain names, and for easily setting up websites – something nearly every student finds interesting and useful.
    - Uniquely, the material includes sections on HTML, CSS, and Javascript, designed for novices, and with built-in hands-on windows for learning and tinkering with each, so that those items are no longer just terms to be remembered, but have concrete meaning to the student who has worked with each.
- Operating Systems
    - OS basics, common systems, being a good OS user, and device drivers. The material includes stories of how popular OS'es came about (and Microsoft's fascinating origins). While OS'es strive to hide details from a user, OS'es aren't perfect; the material thus provides practical advice of how a user can improve efficiency and resolve problems.
- Computer Applications
    - Word processing, spreadsheets, presentation apps, databases, audio and video players, PDF viewer, and audio/video/file compression. The material also teaches students how to troubleshoot common computer hardware or software problems via a methodical approach. The material strives to ensure students know the difference between text, Word, and PDF files, to know how audio and video files work, and to really appreciate the power of compression.
    - Uniquely, the material includes built-in basic word processing and spreadsheet tools. While many computing technology classes have an accompanying lab that teaches such office applications, the included basic tools ensure all students get experience as to those tools' main capabilities and purposes.
- Web / Wireless Apps
    - Video, Wikipedia, social networking, email, text messages, blog, and cloud storage. Clearly, the universe of apps is huge; the chapter touches on a few key ones.

- Privacy
  - Leaving footprints, lack of anonymity, the value of information, how others can listen, and more. This chapter is adapted from excellent open-use privacy content provided by UC Berkeley.
- Security
  - Basics, viruses and malware, account security, Internet scams and spam, cryptography, and denial-of-service attacks. The material focuses heavily on teaching students to be wise and careful computing users.
- More Societal Issues
  - Outsourcing and crowdsourcing, E-commerce, online dating services, health and computer use, and effective email communication. This chapter touches on various topics, and will likely grow. The health topic discusses the latest research on the importance of avoiding sitting for long periods. The unique effective email communication topic provides concrete advice for how to communicate effectively (as well as what to NOT put in email).

A key feature is that the material is modular. As such, instructors can re-order the chapters, move sections within or among chapters, exclude topics, set topics as optional, and more.

Furthermore, the modularity allows for combining with other material. Many instructors have combined some of the above material with other zyBook material on introductory programming (in C, C++, Java, or Python). Some put computing technology topics at the beginning of such a course, to provide a gentle welcoming introduction, before diving into programming concepts. Others put the topics at the end, to help students broaden their perspective beyond programming. For example, we teach an Intro to CS (in C++) course by ending with the above Internet / Web chapter, with students programming some Javascript, thus immediately seeing that programming concepts extend across languages, and seeing how their new programming skills can be immediately applied to create powerful webpages.

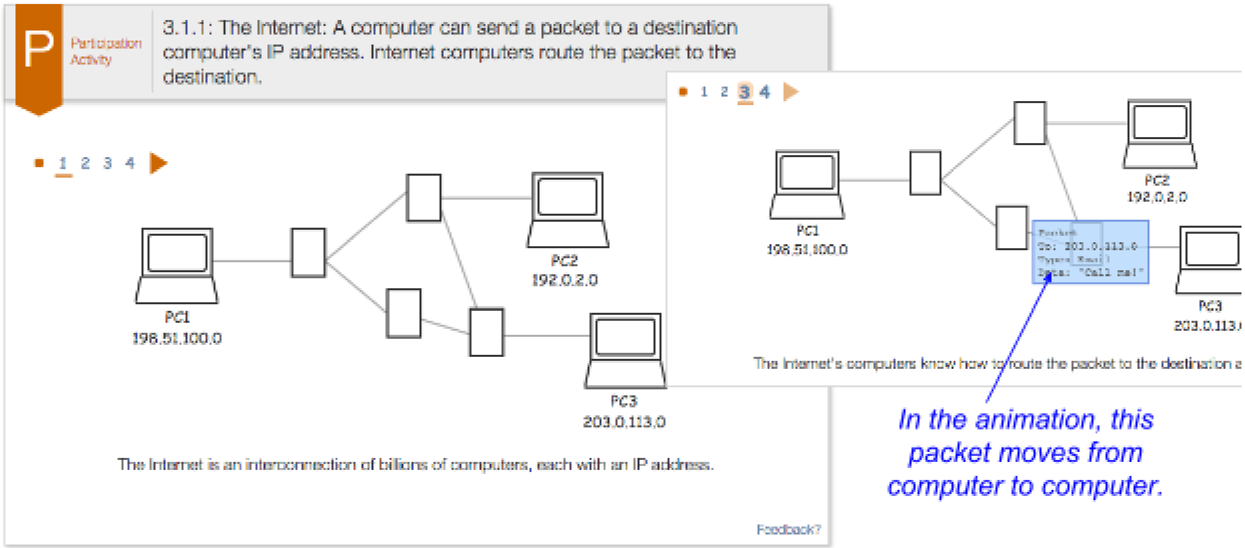
The following sections highlight particular features found throughout the above-listed content.

Animations

Numerous concepts in computing technology have dynamic features, where dynamic means changing in time. For example, the Internet is not merely a static connection of computers, but includes dynamic sending of packets and routing of those packets through intermediate devices.

A CPU is not just a static piece of hardware, but dynamically executes one instruction at a time from a program. The benefit of a larger RAM can be best seen by dynamically showing pieces of programs and data being copied from hard drive, with larger RAM resulting in less copying.

Thus, the new material makes extensive use of animations. Of course, this paper cannot depict an animation dynamically, so uses static snapshots; the reader is asked to imagine how items would be moving in the animation. The image below includes snapshots from an animation that shows packets flowing across the Internet through intermediate computers. The animation includes scenes showing how packets between the same two computers may take different routes, depending on whether intermediate computers are busy.



3.1.1: The Internet: A computer can send a packet to a destination computer's IP address. Internet computers route the packet to the destination.

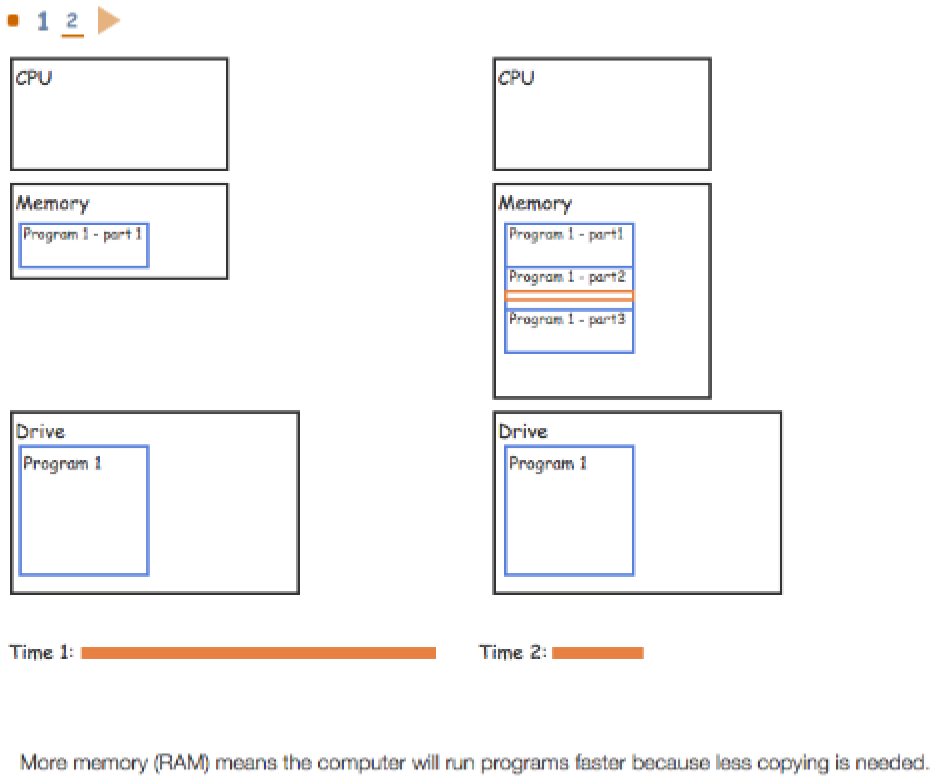
The Internet is an interconnection of billions of computers, each with an IP address.

The Internet's computers know how to route the packet to the destination a

In the animation, this packet moves from computer to computer.

The snapshot below is from an animation showing the CPU executing one instruction at a time, with a user typing on a keyboard, causing data to be transferred from the keyboard to memory, and then to the screen.

The snapshot below is from an animation showing a program executing, copying program parts into memory to execute. The student feels the speedup as the animation runs faster for the larger memory on the right than for the smaller memory on the left; the time bars grow as the animation runs to statically represent time required. After watching the animation, the static representation of the figure also makes more sense to the student, who saw that the time bars represent execution time.

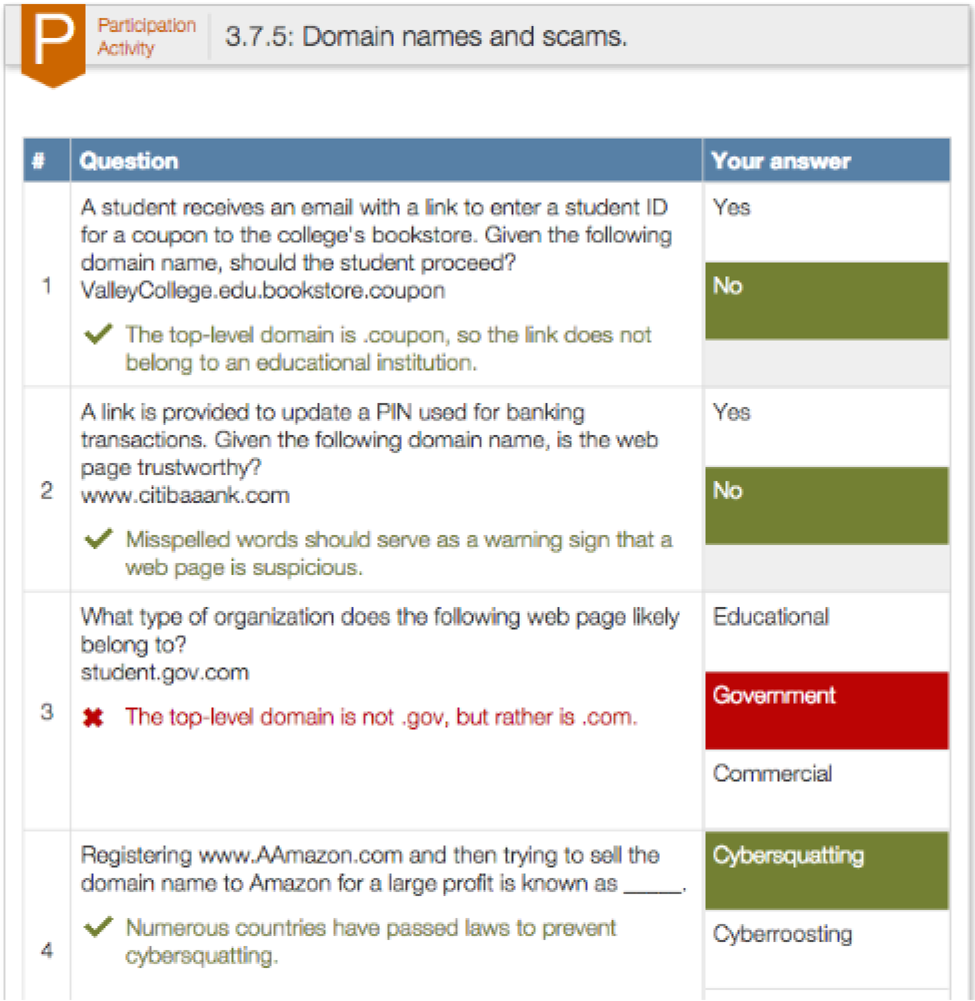


More memory (RAM) means the computer will run programs faster because less copying is needed.

Learning questions

The material makes extensive use of learning questions. Rather than pages of text and/or figures, the material uses less text/figures and then continues teaching through interactive questions (true/false, multiple choice, short answer, and matching). Explanations of correct answers provide further information, and explanations of incorrect answers explicitly break down common misconceptions. This style of material is akin to engaging the student in a dialog, rather than lecturing to the student.

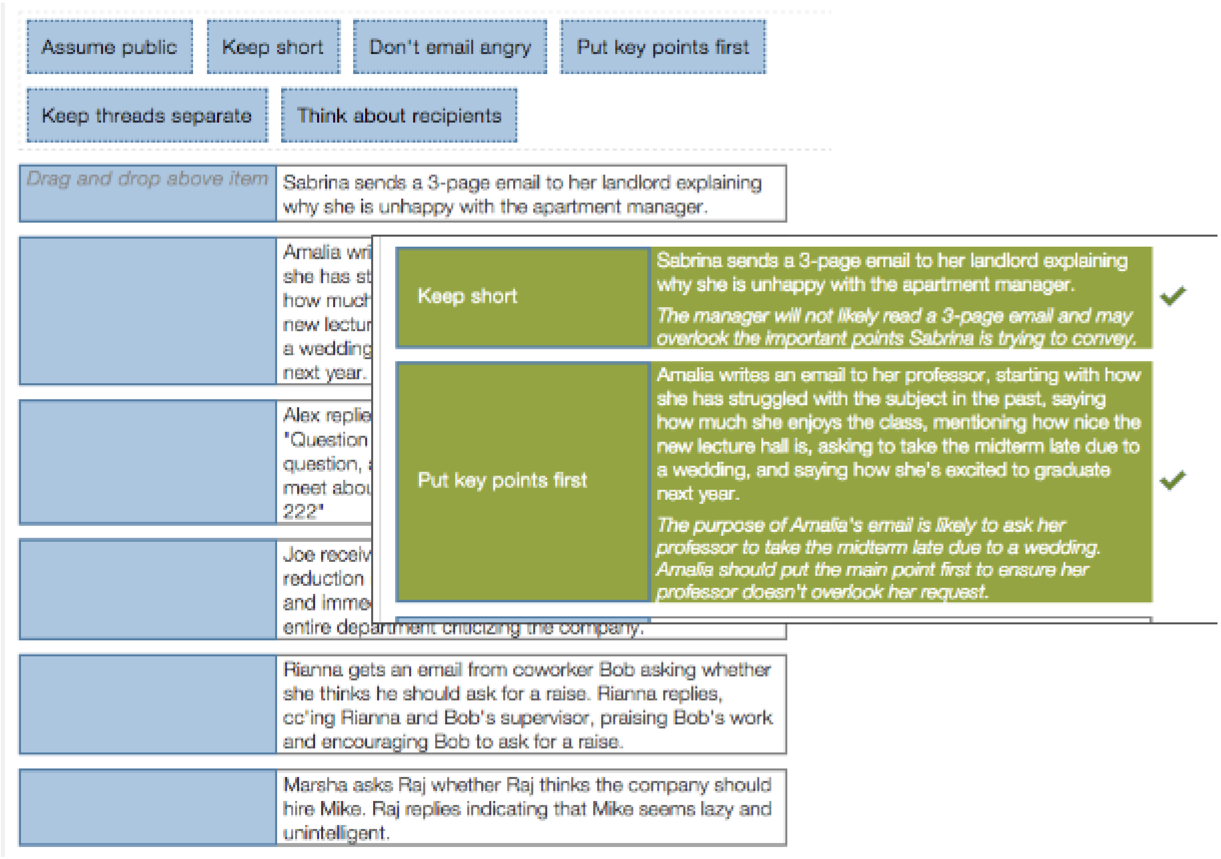
The following image depicts a typical set of learning questions. Notice that the wrongly-answered question gives the student a real experience in being tricked by a misleading domain name, so the student hopefully will not be similarly tricked. Such tricking is common throughout the material, providing "learning moments". Note also that the correct answers have explanations that clarify or elucidate.



P Participation Activity 3.7.5: Domain names and scams.

#	Question	Your answer
1	A student receives an email with a link to enter a student ID for a coupon to the college's bookstore. Given the following domain name, should the student proceed? ValleyCollege.edu.bookstore.coupon ✓ The top-level domain is .coupon, so the link does not belong to an educational institution.	Yes 🟩 No
2	A link is provided to update a PIN used for banking transactions. Given the following domain name, is the web page trustworthy? www.citibaaank.com ✓ Misspelled words should serve as a warning sign that a web page is suspicious.	Yes 🟩 No
3	What type of organization does the following web page likely belong to? student.gov.com ✖ The top-level domain is not .gov, but rather is .com.	Educational 🟥 Government Commercial
4	Registering www.AAmazon.com and then trying to sell the domain name to Amazon for a large profit is known as _____. ✓ Numerous countries have passed laws to prevent cybersquatting.	🟩 Cybersquatting Cyberroosting

The following image depicts a matching learning question. In a normal textbook, the provided examples would have been a table or a bulleted list. Below, the student is engaged in a more active process, requiring thought to correctly match. For correct matches, an explanation describes why the match is correct. Learning questions are not quizzes or homework, but rather a more engaging way of reading, and hence explanations are central and thus always present.



Sabrina sends a 3-page email to her landlord explaining why she is unhappy with the apartment manager.

Amalia wri
she has st
how much
new lectur
a wedding
next year.

Alex replie
"Question
question,
meet abou
222"

Joe receiv
reduction
and imme
entire department criticizing the company.

Rianna gets an email from coworker Bob asking whether she thinks he should ask for a raise. Rianna replies, cc'ing Rianna and Bob's supervisor, praising Bob's work and encouraging Bob to ask for a raise.

Marsha asks Raj whether Raj thinks the company should hire Mike. Raj replies indicating that Mike seems lazy and unintelligent.

Sabrina sends a 3-page email to her landlord explaining why she is unhappy with the apartment manager.

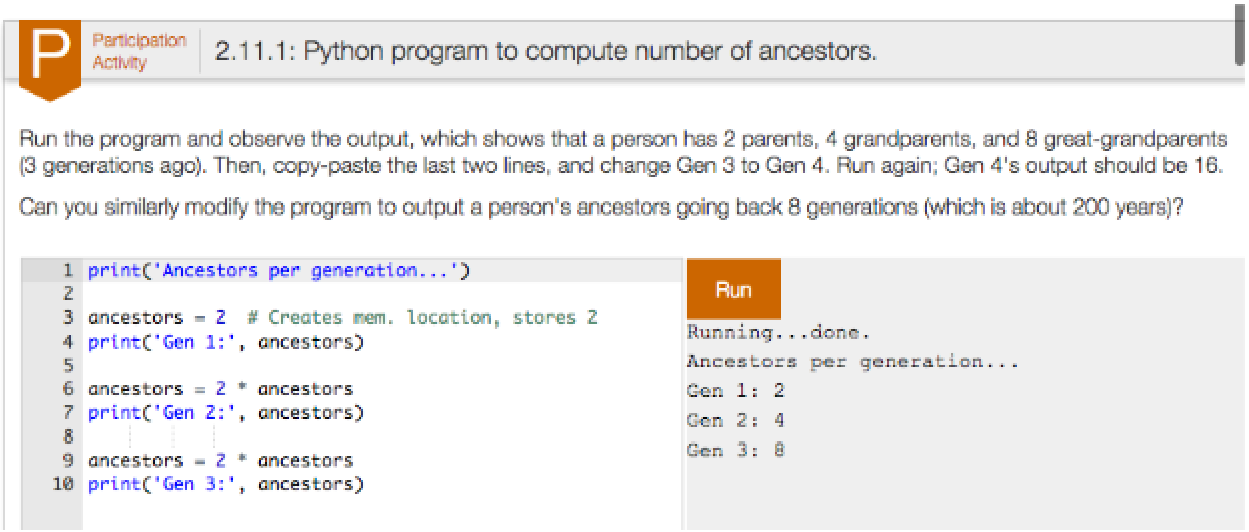
The manager will not likely read a 3-page email and may overlook the important points Sabrina is trying to convey. ✓

Amalia writes an email to her professor, starting with how she has struggled with the subject in the past, saying how much she enjoys the class, mentioning how nice the new lecture hall is, asking to take the midterm late due to a wedding, and saying how she's excited to graduate next year.

The purpose of Amalia's email is likely to ask her professor to take the midterm late due to a wedding. Amalia should put the main point first to ensure her professor doesn't overlook her request. ✓

Integrated Python environment

Many instructors of a computing technology course try to give students a simple programming experience, so that the student can proudly know they have written at least one computer program in their lives, even if a simple program. However, setting up a programming environment like Python, though relatively simple for a tech savvy person, can overwhelm students in a computing technology class. Fortunately, because our material is entirely web-based, the material can include a Python programming window directly within itself. Behind the scenes, the window interfaces with an AWS (Amazon Web Services) server, of which the student is unaware. The student merely clicks "Run" to run the default program, and then can make modifications to that program and run again.



2.11.1: Python program to compute number of ancestors.

Run the program and observe the output, which shows that a person has 2 parents, 4 grandparents, and 8 great-grandparents (3 generations ago). Then, copy-paste the last two lines, and change Gen 3 to Gen 4. Run again; Gen 4's output should be 16.

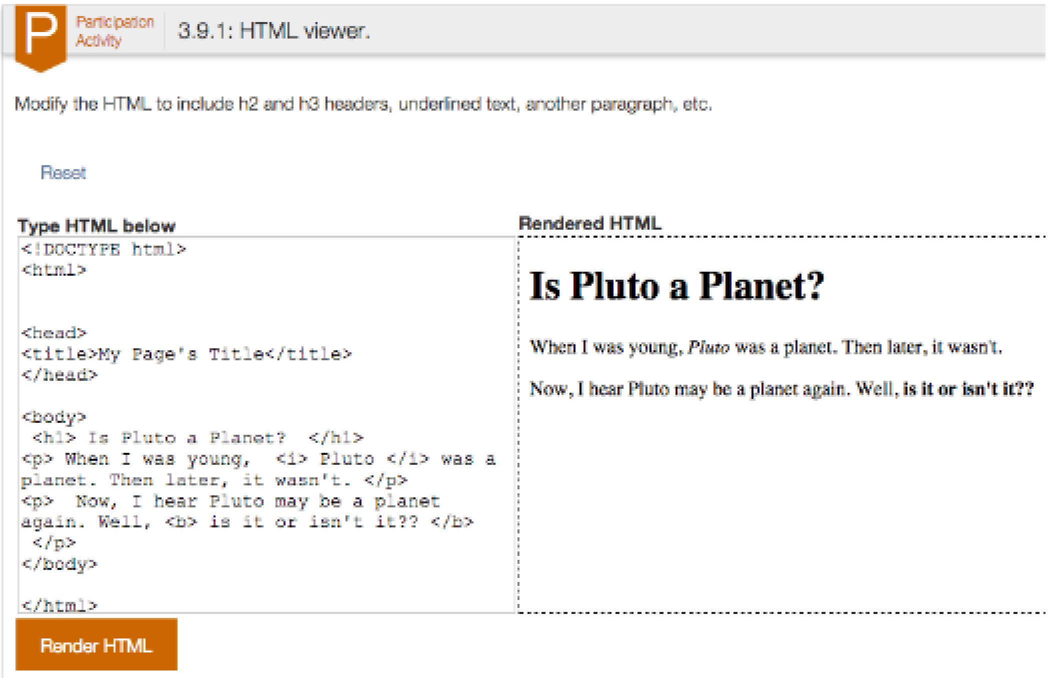
Can you similarly modify the program to output a person's ancestors going back 8 generations (which is about 200 years)?

```
print('Ancestors per generation...')

ancestors = 2  # Creates mem. location, stores 2
print('Gen 1:', ancestors)

ancestors = 2 * ancestors
print('Gen 2:', ancestors)

ancestors = 2 * ancestors
print('Gen 3:', ancestors)
```

```
Running...done.
Ancestors per generation...
Gen 1: 2
Gen 2: 4
Gen 3: 8
```

Integrated HTML, CSS, and Javascript tools

Among the most interesting and relevant computing technology topics for students relate to modern web technologies. However, most textbooks can only provide static figures trying to depict each of these technologies, which is hard for students to really understand. Fortunately, because our material is entirely web-based, the material can integrate tools so that students can execute each of the above languages and immediately see the impact on a real web page. For example, the below image shows a tool that provides HTML on the left, and the rendered web page on the right. The student can change the HTML, by modifying text, adding bold, inserting header text, and more, and then press "Render HTML" to see how the changes are reflected in a new webpage.



3.9.1: HTML viewer.

Modify the HTML to include h2 and h3 headers, underlined text, another paragraph, etc.

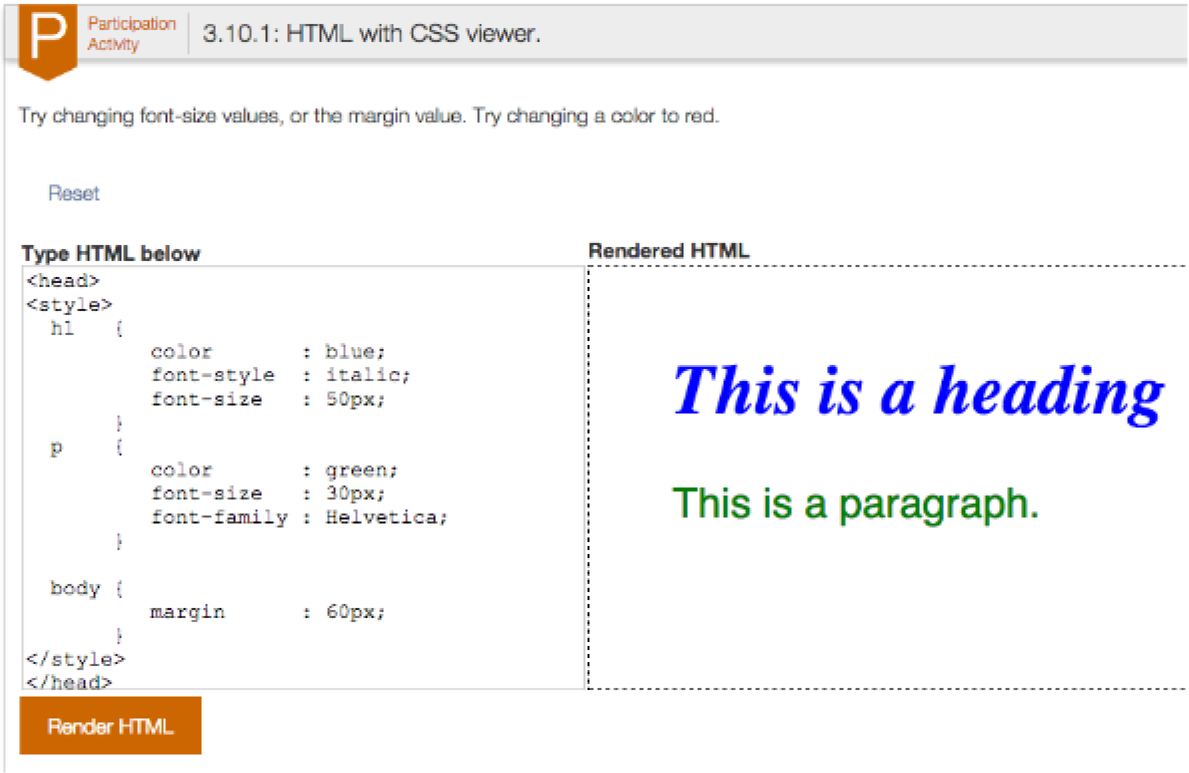
```
<!DOCTYPE html>
<html>


<head>
<title>My Page's Title</title>
</head>

<body>
 <h1> Is Pluto a Planet?  </h1>
<p> When I was young,  <i> Pluto </i> was a
planet. Then later, it wasn't. </p>
<p>  Now, I hear Pluto may be a planet
again. Well, <b> is it or isn't it?? </b>
 </p>
</body>

</html>
```

Is Pluto a Planet?

When I was young, *Pluto* was a planet. Then later, it wasn't.

Now, I hear Pluto may be a planet again. Well, **is it or isn't it??**

Subsequent images show similar tools for CSS and Javascript.



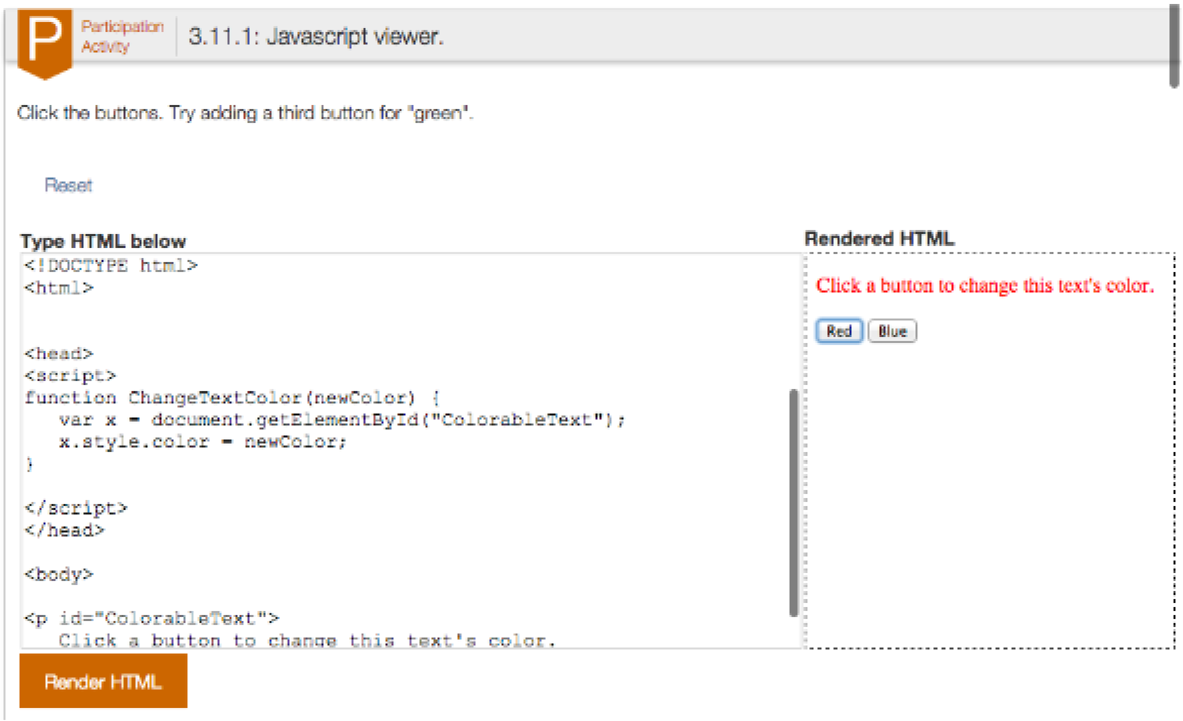
3.10.1: HTML with CSS viewer.

Try changing font-size values, or the margin value. Try changing a color to red.

```
<head>
<style>
  h1    {
          color        : blue;
          font-style   : italic;
          font-size    : 50px;
        }
  p     {
          color        : green;
          font-size    : 30px;
          font-family  : Helvetica;
        }

  body  {
          margin       : 60px;
        }
</style>
</head>
```

This is a heading

This is a paragraph.



3.11.1: Javascript viewer.

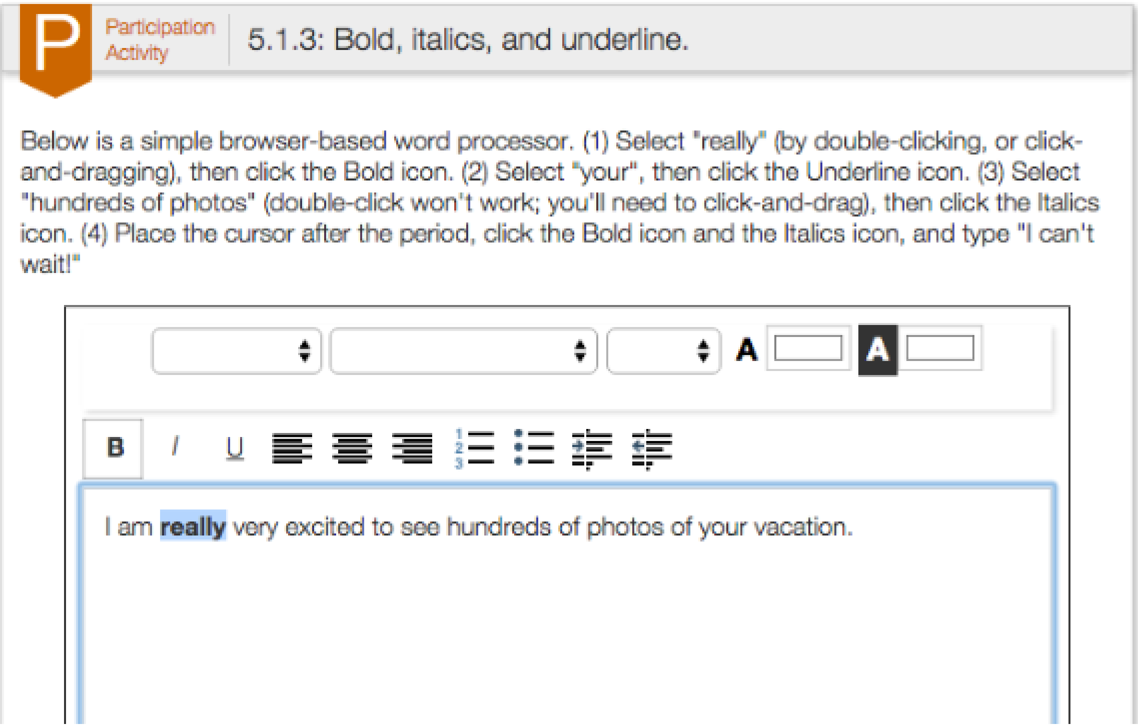
Click the buttons. Try adding a third button for "green".

```
<!DOCTYPE html>
<html>


<head>
<script>
function ChangeTextColor(newColor) {
   var x = document.getElementById("ColorableText");
   x.style.color = newColor;
}

</script>
</head>

<body>

<p id="ColorableText">
   Click a button to change this text's color.
```

Click a button to change this text's color.

Integrated basic word processing and spreadsheet applications

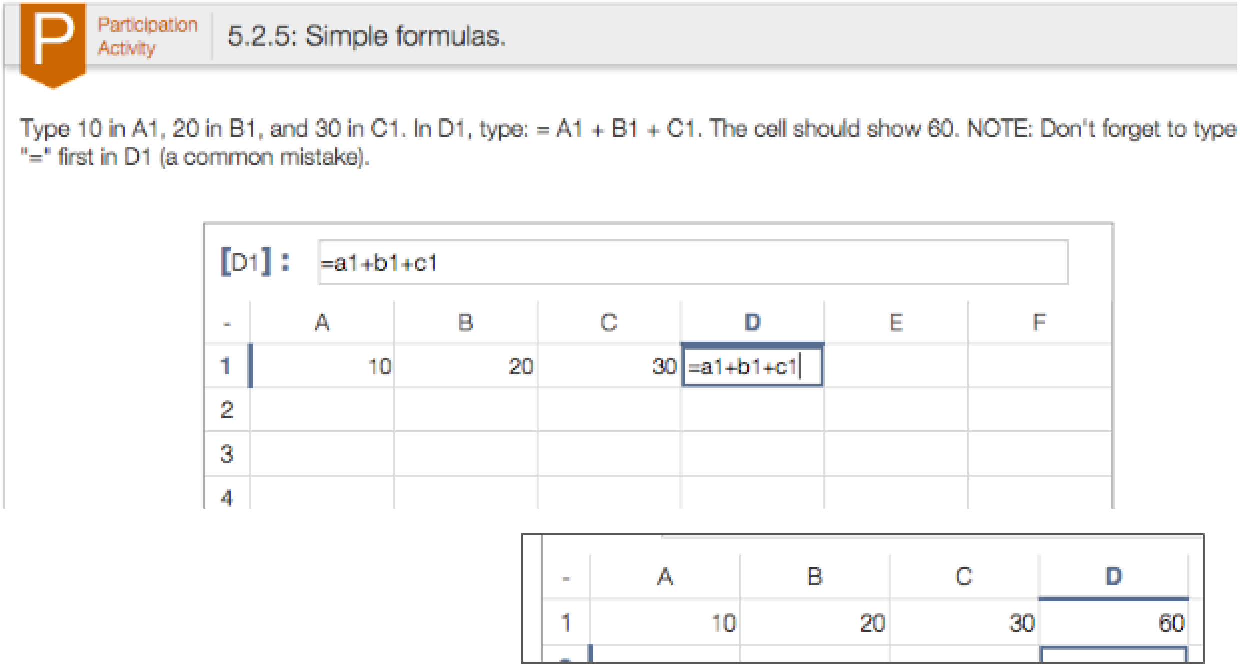
Many computing technology courses have lab components that teach office applications like word processing, spreadsheets, and presentation apps. However, some courses do not have such labs. As such, textbook coverage of those apps have a hard time conveying the concepts, such as how to change the font for a portion of text, or how to use formulas in a spreadsheet. Even with a lab, having a simple introduction to those apps can help students quickly grasp basic concepts, freed from distracting details of running full office apps. Our material includes built-in basic versions of word processing and spreadsheet apps, as depicted below. Students can explore basic office app concepts without leaving the web browser.



5.1.3: Bold, italics, and underline.

Below is a simple browser-based word processor. (1) Select "really" (by double-clicking, or click-and-dragging), then click the Bold icon. (2) Select "your", then click the Underline icon. (3) Select "hundreds of photos" (double-click won't work; you'll need to click-and-drag), then click the Italics icon. (4) Place the cursor after the period, click the Bold icon and the Italics icon, and type "I can't wait!"

I am **really** very excited to see hundreds of photos of your vacation.



5.2.5: Simple formulas.

Type 10 in A1, 20 in B1, and 30 in C1. In D1, type: = A1 + B1 + C1. The cell should show 60. NOTE: Don't forget to type "=" first in D1 (a common mistake).

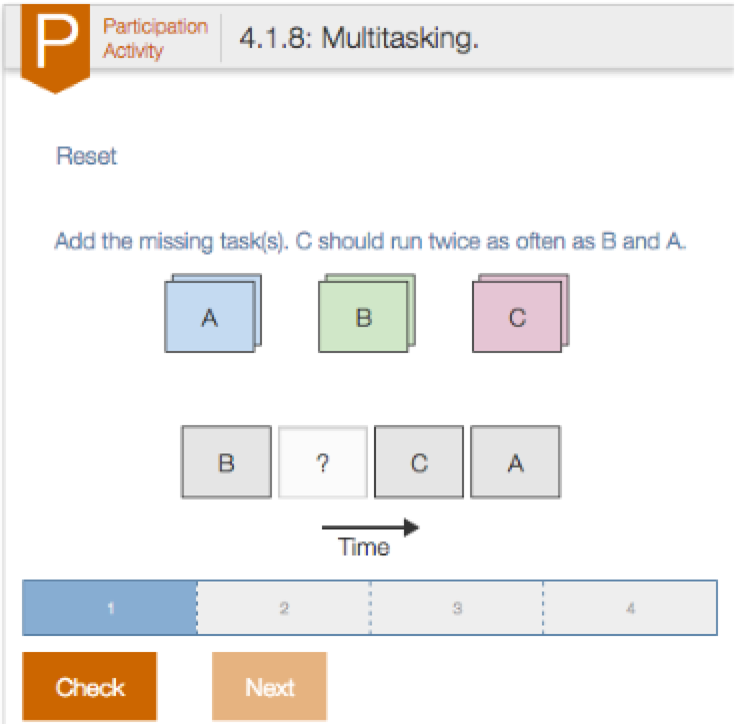
[D1] : =a1+b1+c1

-	A	B	C	D	E	F
1	10	20	30	=a1+b1+c1		
2						
3						
4						

-	A	B	C	D
1	10	20	30	60

Other activities

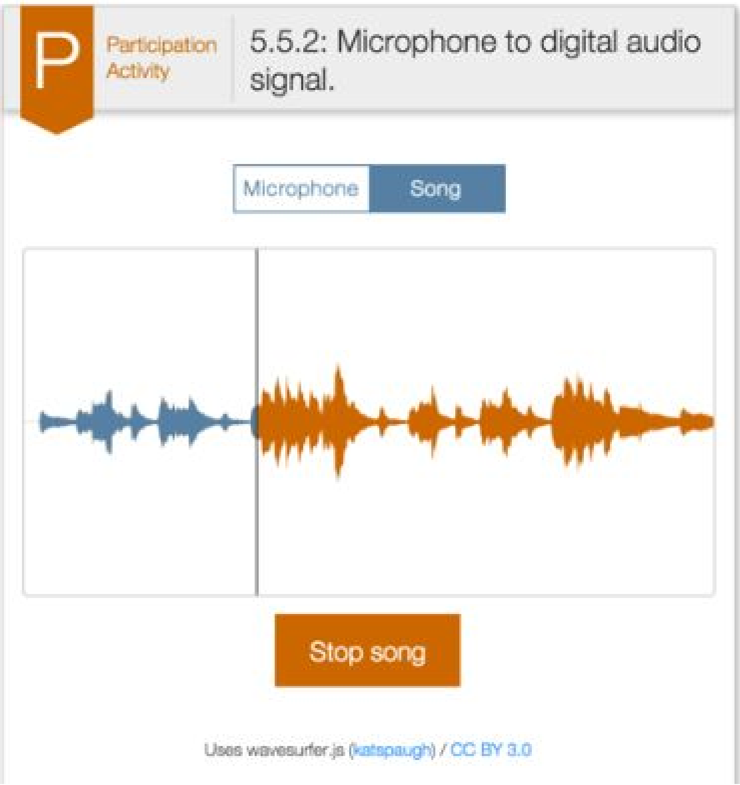
Numerous other activities exist to engage the student and help build understanding. For example, after explaining how an operating system carries out multitasking via scheduling, the student is asked to act like an OS and schedule tasks.



4.1.8: Multitasking.

Add the missing task(s). C should run twice as often as B and A.

As another example, when the student is taught how audio signals are converted to digital values, a tool allows the student to view the signals generated from the student's computer microphone.



5.5.2: Microphone to digital audio signal.

Uses wavesurfer.js (katspaugh) / CC BY 3.0

Numerous other examples of activities exist. Some activities make use of existing websites. For example, one activity has the student use an existing speedtest website to determine the student's actual Internet download and upload speed. Another provides information on the student's IP address and shows the student's location on a map. Another allows a student to upload a photo taken with a smartphone and then to see the actual GPS information embedded in that photo. These activities bring the material to life for the student, making the concepts concrete and relevant to the student's perspective.

Cross-semester comparison: Students significantly prefer animated and interactive content

We sought to measure student preference for using a static textbook or for using an animated and interactive web-based learning content in a course. We surveyed 1,097 students from the same course (CS8: Introduction to Computing) at the University of California at Riverside, across two offerings: 563 students in Spring 2015 using a static textbook and 534 students in Fall 2015 using animated and interactive web-based learning content. Both offerings had the same instructor. The students were blind to the conditions of the survey.

The survey included agreeability statements on a 6-point scale: 6 = Strongly agree and 0 = Strongly disagree. Students with the web-based content reported a 5.4 for "The textbook contributed to my success in the course.", which was significantly higher (p-value < 0.001) than the 4.4 reported by students using a static textbook. Similarly, students using the web-based content reported a 5.3 for "When reading the textbook, I usually was able to understand the concepts being taught.", which was significantly higher (p-value < 0.001) than the 4.6 reported by students using a static textbook.

The students' ability to better understand the animated and interactive web-based content compared to the static textbook content likely led to the feeling that the web-based content contributed more to the success in the course than the static textbook.

Conclusions

Computing technology is one of the most interesting, relevant, and useful subjects taught to college students, as well as high-school students and adult learners. Most learning materials are rooted in textbooks, where only text and figures were available to teach concepts. As such, most books over-emphasize terms and definitions, especially books in their Nth edition (e.g., 15th edition) which have become bloated with new terms. Some new books take a more visual approach, with more photos and less text, but are still rooted in static textbooks. By breaking from textbooks entirely, we were able to develop material that emphasizes learning through animations, interactive learning questions, embedded tools for hands-on experiences (Python, HTML, CSS, Javascript, word processing, spreadsheets, and more), and numerous miscellaneous activities to make concepts concrete and often personally relevant (Internet speed test, smartphone photo GPS extractor, etc.). The first version of the material was released in Summer 2015, and already has been used by over 20 universities, with continuing rapid growth. By the nature of computing technology, the material will continue to be updated (made much easier due to being entirely web-based).

Acknowledgements

This work was supported in part by the National Science Foundation's Small Business Innovative Research (SBIR) program (1315094 and 1430537), and a Google faculty research award. We are sincerely grateful for their support.