Computer Science Everywhere

More nonmajors are delving deep into the computer science curriculum to prepare for work and life in the 21st century

Computer science is having a moment, and Northwestern students know it. Even during their short collegiate careers they have felt the field shift from unapproachable to accessible, from nerdy to cool, from fluorescent-lit basement labs to glamorous, trendy workplaces.

“When you can program, people look at you like you’re a wizard,” says Daniel Learner, a theater and psychology major who didn’t take the computer programming class offered at his high school because only the “really, really, really nerdy kids took it.” Now Learner, a senior, typifies many of the students found in McCormick’s computer science courses: nonmajors who understand that computer science is a gateway to tomorrow’s careers.

Over the past five years enrollment in undergraduate computer science courses has tripled at Northwestern. While the number of computer science majors has nearly doubled, many students enrolled in the courses are nonmajors looking to enhance their skill sets. McCormick has responded to demand by allowing more students into classes and offering more options. Classes like Data Structures that were previously offered once a year are now offered three or four times.

“If you go back several years it might be that students were taking Introduction to Programming to gain some computer literacy,” says Professor Alan Sahakian, chair of the electrical engineering and computer science department. “More recently what I’m seeing is nonmajors taking courses deep into the computer science curriculum—courses that, in the past, only majors would have taken.”

Daniel Learner, a theater and psychology major, wrote a program that compared headlines from Northwestern’s two student publications. He sent it to a friend, who wrote that Learner was “a magical human being.”
“In computer science there are rules and syntax, but within that framework there is so much room for creativity.”

KATIE ZHU

The shift can be attributed in part to the explosion of computing in our everyday lives. An influx of data in industries as varied as healthcare and manufacturing has left the market thirsty for employees who have the computing knowledge to make sense of it all. And newly minted college graduates, who grew up during the advent and proliferation of social media, enter the job market in a world where developing the right app could make them billionaires—or, at least, highly marketable to employers.

“Computer science is becoming more like a basic science, like mathematics or physics,” Sahakian says. “More and more commonly, students are asked in interviews whether they know how to program Java or Ruby on Rails. Employers want students with that experience.”

**A trifecta of studies**

Learner didn’t have his career in mind when he took Introduction to Computer Programming after a student in his theater class recommended it. “It just sounded cool,” he says. He learned to program in C++, and he was hooked. In his next computer science class he wrote a program that compared headlines from two student publications—the *Daily Northwestern* and *North by Northwestern*—and found the words they had in common. It took only an hour and a half to write the program.

“I sent it off to my friend, who said, ‘You’re a magical human being,’” Learner says. “It was a class where you go in, learn stuff, and leave with a usable skill.”

Learner isn’t interested in majoring in computer science—two majors are enough for him—but he is applying what he’s learned to other fields. In theater lighting design, for example, the equipment uses simple computers that run on a hard-to-use language. “The language is confusing and difficult, and it shouldn’t be. It’s become a pet project,” he says.

He also used his programming skills in a psychology research methods course. While other students sent out surveys to their friends and begged for responses, Learner wrote a program that mined Twitter for smiling and frowning emoticons.

“I used that to determine how happy or sad people were throughout the day,” he says. “Where some students had 20 or 30 responses, I had 20,000 pieces of data.” His work at the intersection of psychology and computer science helped him land a summer internship with advertising agency Digitas, where he worked on digital strategy for online advertising.

“I like it when my academic interests intersect,” he says. “When it happens I get to do cool things.”

Katie Zhu, a journalism major, began to study computer science to learn new skills. She won the AP-Google Journalism Technology Scholarship to create an open-source portal to facilitate collaboration among journalists.

**Taking technology to media**

That intersection entices Katie Zhu as well. Zhu arrived at the Medill School of Journalism, Media, Integrated Marketing Communications aiming to be the next Bob Woodward. But then she found that it wasn’t the articles in student magazine *North by Northwestern* that intrigued her—it was the publication’s online housing guide, a Flash graphic that let users decide on housing options based on room sizes and amenities. Zhu joined the magazine’s interactive desk
her freshman year and began learning how to program basic interactive graphics.

“I remember the first time I made an alert in Javascript,” she says. “I made a dialog box pop up. I remember thinking that was so cool, that you could tell computers what to do.”

She worked with Jeremy Gilbert, associate professor of journalism, on a program that rewarded users for reading news articles. It was then that she realized her technical skills were lacking. “I didn’t have any web development experience,” she says. “I didn’t really know how the Internet worked. I didn’t know what a server was.”

But it was the movie The Social Network—especially the rhythmic scene in which, after getting jilted by his girlfriend, Mark Zuckerberg builds a website that ultimately crashes Harvard University’s servers—that inspired Zhu to pursue a second major in computer science.

“College is about finding your true nature,” she says. “And though seeing my byline in a newspaper was rewarding, after I’d spent two hours debugging some code only to find out that I’d misspelled some variable—a very small, dumb mistake—and then working to finally get that code to run, that level of satisfaction was more fulfilling for me.”

Zhu admits to taking the hard road—she has never been a math and science person, she says, and she didn’t have a predisposition to programming—but she has found the resources and put in the time to develop a deep technical knowledge base.

Her efforts were rewarded last year, when she was one of six students nationwide to win the AP-Google Journalism and Technology Scholarship. In her application Zhu submitted a proposal for LedeHub, an open-source tool that would allow journalists and developers in the newsroom to share work and collaborate on projects. (“Lede” is a journalism term for the first paragraph of an article.) She’s refining the idea after an internship at NPR last summer showed her the need for early-stage collaboration among reporters, editors, producers, and programmers.

After graduating in June, Zhu spent the summer interning on the New York Times’s interactive desk before joining Medium, a content-creation platform started by Twitter cofounders Evan Williams and Biz Stone. The job provides her with a way to help solve problems at the intersection of technology and media.

“Computer science teaches you a brand-new way of thinking, a new way to approach problems,” she says. “It’s scientific—there are rules and syntax—but within that framework, there is so much room for creativity.”

Programming by day, making music by night

Creativity mixed with rules and syntax is the recipe for another field of study: music. No one knows this better than Lee Fan, a viola player who double-majored in music performance and computer engineering.

“Actually, I really wanted to go into just music, but my mom told me that if I wanted to do music, I had to have another degree,” he says. Programming came naturally to Fan after he taught himself to program basic games on his Texas Instruments calculator in middle school. At the same time his music teacher convinced him to leave the overcrowded field of violinists to play the deeper-voiced, and less commonly studied, viola.

“I like making music,” he says. “I appreciate it as art and entertainment.” His high school years were spent playing the viola in several orchestras and reading books on C++ that were lying around his house.

By the time he applied to Northwestern, his top school choice for its combination of music and engineering programs, Fan was more interested in how the computer itself interacted with programs. “I just could not wrap my head around how zeroes and ones would be interpreted by a computer,” he says, so he chose computer engineering for his second major.

It turned out that writing code came more naturally than working with hardware. Fan found himself taking more and more computer science courses during the day and practicing his viola at night. To him, they were two separate worlds. In computer science, once he learned a piece of knowledge, he retained it without effort; it was digital, nondegradable. But his viola was analog: he had to constantly practice or the information would degrade—quickly. “If I don’t touch my instrument for a day, I’m out of tune,” he says.

The only crossover is technique. Before he plays, Fan imagines what the music should sound like. He uses a similar strategy when he
codes. “Many people just like to jump in and start coding, but I like to play it out in my head. I’ll run simulations and debug mentally before I write anything.” He’s also channeled his creative side into developing a crowd-sourcing storytelling program that allows users to collaborate on choose-your-own-adventure narratives.

Fan decided to stay on at Northwestern for an extra year to earn his master’s in computer science through McCormick’s BS/MS program. After completing his degree he hopes to find the right balance of working for a large software company and teaching viola. “I always thought I would end up playing music and writing code,” he says.

“A lot of fun and more interesting”
Sometimes nonmajors who take computer science courses don’t come all the way from across campus: they inch over from within the same department. That is the case for Gabriel Peal, a 2013 electrical engineering alumnus who took most of the computer science major courses.

“I almost molded it into a computer science degree,” he says. “That it wasn’t is more of a technicality.” Initially drawn to electrical engineering’s curriculum that teaches students the theory behind circuits and hardware but leaves plenty of quarters open for classes in other disciplines, Peal quickly learned that computer science “was a lot of fun and more interesting to me. I looked at what companies I might work for—Google, Apple—and they were just incredible.”

Peal was no stranger to coding. He had developed a program on his TI-83 calculator, one that could solve all the problems in his high school geometry class. But he thought computer science was for nerds. “I had this conception of it being bland guys in button-down shirts sitting in cubicles and typing code all day. I was really glad to discover through my internships that it wasn’t the case.”

What sealed Peal’s fate was the NUvention: Web course, where cross-disciplinary student teams are tasked with developing a web or software business. Peal’s team developed and launched Stagecoach, a project management solution for the film industry. Though the team didn’t continue developing the business after the class ended, the experience remains one of the most important of Peal’s undergraduate career.

“It forced me to learn new web and app development tools—and to launch an actual business,” he says. “It really made me grow as a software engineer. We had to work in a team, set goals, and achieve them in a timely manner. It opened my world to so many possibilities. It was hands-down the best course I took.”

Peal had an internship at Google and ultimately accepted a position as a software engineer on the company’s Android team. “It’s a collaborative work environment, an open atmosphere,” he says. “Everyone there is at such a high level.” That’s a far cry from what a high-school-aged Peal had imagined life as a software engineer would be.

“The opportunities to change the world through computer science are available to anybody,” he says. “We get to change people’s lives and do really fun things.”