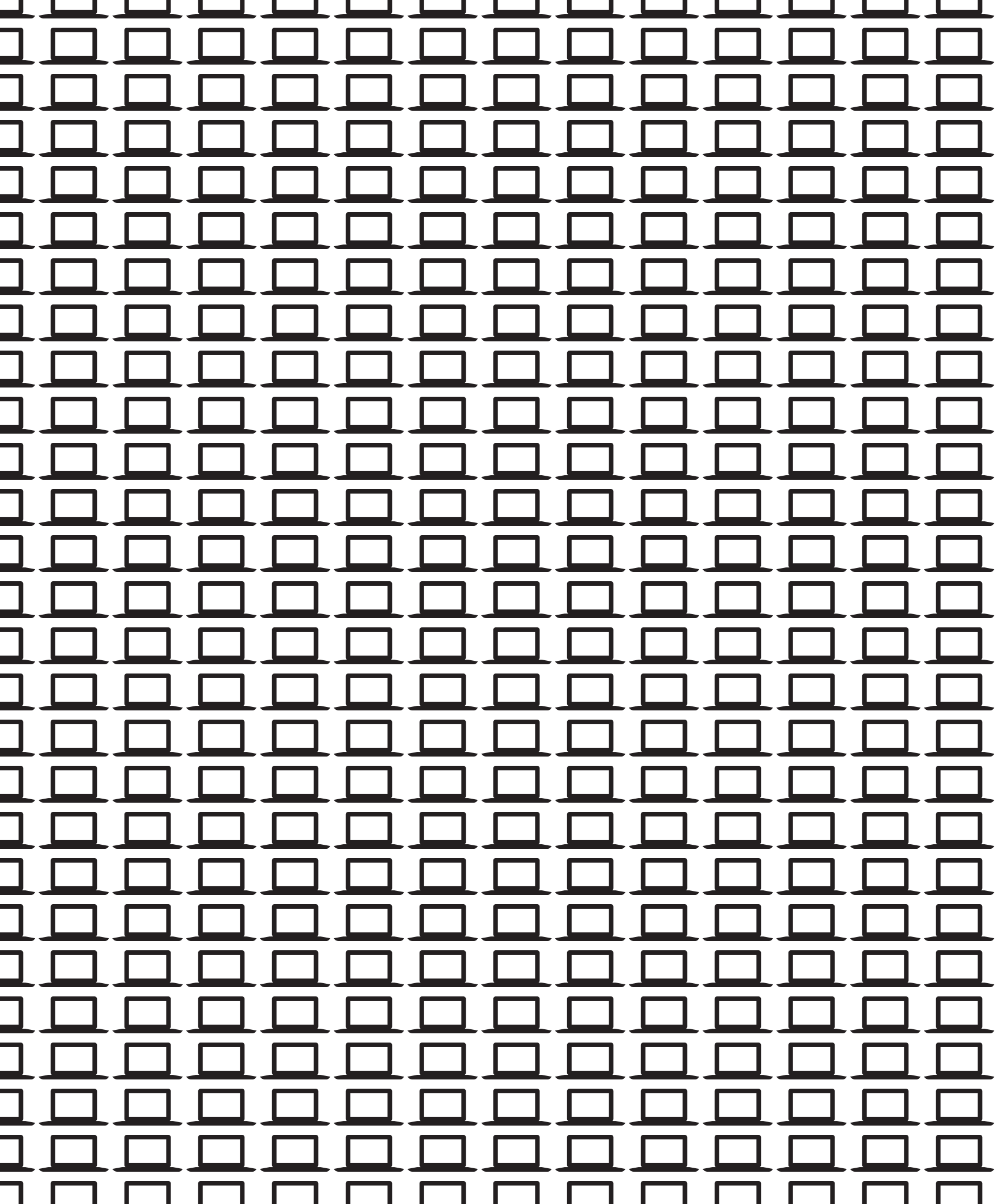

COMPUTER SCIENCE FOR EVERYONE. IF YOU CAN FIND A SEAT.

THE MORE TECHNOLOGY BECOMES EMBEDDED IN DAILY LIFE, THE GREATER THE DEMAND FOR COMPUTER SCIENCE COURSES.

Standing room only. That's a phrase not often associated with a computer science class, but it describes what's on the verge of happening in Northwestern Engineering's popular computer science course CS 111: Fundamentals of Computer Programming.

"THE EXCITEMENT AND DEMAND FOR COMPUTER SCIENCE FROM OUR STUDENTS INSPIRES US TO IMAGINE NEW WAYS OF TEACHING AND LEARNING BOTH IN THE CLASSROOM AND BEYOND."

LARRY BIRNBAUM PROFESSOR OF COMPUTER SCIENCE

And this is not your typical classroom. CS 111 is held in Ryan Family Auditorium in the Technological Institute, a space usually reserved for large events like commencement, speeches, and even small concerts. But it is the only space large enough to accommodate all the students who want to enroll in this course.

The influence of computer technology in virtually every aspect of daily human life has created a swelling demand for computer science courses everywhere. At Northwestern alone, the number of computer science majors has exploded—tripling in just the past five years. And it's not only the majors who want to learn about the burgeoning field. CS 111 is Northwestern Engineering's most popular class for non-majors, who come from nearly every school within the University.

CS 111 and another popular course, CS 101: Computer Science for Everyone, are spreading the subject to all types of engineers and beyond. Both classes are open to all Northwestern majors, have no prerequisites, and are accessible to students of all skill levels. The classes provide a critical introduction to a powerful field.

"The excitement and demand for computer science from our students inspires us to imagine new ways of teaching and learning both in the classroom and beyond," says Larry Birnbaum, professor of computer science and director of the division. "Computer science has the potential for collaborating with nearly every discipline at Northwestern."

PROGRAMMING FOR EVERYONE

With the influx of value-laden big data that companies now have available, employees with programming skills are hotly sought after to find meaningful insights and trends within the slog of numbers. This is true across practically every field, from sociology and linguistics to finance and athletics. Entrepreneurs hoping to launch the next big app or Internet startup also find enormous advantages in learning how to program. And students from all backgrounds see value in having this all-important skill in their back pockets to develop new and exciting projects on a whim.

"Computer science allows you to build anything," said freshman Anna Furlong, who studies math and computer science. "You can come up with an idea, spend an afternoon coding, and have a prototype quickly. Anything is possible."

To keep up with the ever-changing nature of computer science and to satisfy the booming interest in programming, professors Ian Horswill and Sara Sood revamped CS 111 in the fall of 2015. This included the decision to teach with Racket, a general-purpose programming language compatible with both Windows and Mac OS. Students in the class use the language to design simple tools and games as well as practice diagnostics to find and fix bugs hidden within the code.

But Horswill emphasizes that it is not enough to just write code that works. Students need to learn how to write code that is clear, concise, and elegant and in a way that humans can understand. For this, they need familiarity with human psychology.

"I wanted to make a course that was about ideas rather than punctuation marks," he says. "A lot of software development is about human psychology. But that often gets lost in the rush to explain all the punctuation marks."

PHILOSOPHY FOR EVERYONE

With philosophy as its soul, CS 101: Computer Science for Everyone defies the misconception that computer science is solely about coding. Offered every fall, the class introduces both majors and non-majors to the core ideas that drive the field and how computer science interacts with everything else in the world.

WHAT'S THE RACKET?

Last year, CS 111 instructors started teaching the course with Racket, a general-purpose programming language in the Scheme family. Northwestern Engineering Professor Robert Findler co-developed both Racket and Scheme, the programming language previously used in the class. Here are some reasons why Racket works well in the classroom:

- Racket is naturally cross-platform, performing well in Windows, Mac OS, and Unix.
- It was designed with beginning programmers in mind.
- It is not restricted to a specific programming style.
- It features an extensive library that is thorough and approachable.
- It can be used to create new languages.
- Its interactive mode encourages experimentation.
- It is open-source, so it may be subsequently improved by outside developers.
- Racket is equally unknown. Most students in the class will start at a fair and equal point.

"Computer science can expand to include anything," says Jason Hartline, associate professor of computer science, who teaches the course. "It can help us better understand the world around us."

Each week, the course features different guest professors from the department who present the computer science topics in which they conduct research. Birnbaum, for example, presents on artificial intelligence and machine learning; Fabián Bustamante lectures about systems and networks; and Brenna Argall talks about robotics. These guest lectures change regularly to keep the content fresh.

There are no textbooks, as Hartline prefers to keep class discussions current and sometimes even ripped from the headlines. When Facebook was outed for using an algorithm to customize users' news feeds in order to control their moods, for example, Hartline brought the discussion to class. He also uses TED Talks and articles from popular media.

BEYOND PROGRAMMING

"You learn about topics on a broad scale," says Alec Reinke, a sophomore in mechanical engineering. "I was surprised to learn how many different fields of computer science there are."

This broad approach is by design. When Hartline redesigned the course four years ago, he wanted to give students a deeper understanding of computer science beyond programming. He has long noticed how computer science touches unexpected areas of life, and looks for ways to use it as a tool to understand complex environments. His own research introduces design and analysis methodologies from computer science to understand and improve outcomes of economic systems. In Hartline's world, economic systems work like computers, and economic factors, such as a country's gross domestic product, are the computation's outcome.

Historically, most introductory classes have focused on coding. While it's undoubtedly a critical skill, programming is only a portion of computer science. Hartline likens it to a high school biology course.

"Biology classes aren't just centered around one skill like dissecting a frog," Hartline says. "Students don't go to class every day to learn more specialized methods to get better and better at dissecting. Computer science also has very many advanced topics, such as cryptography and machine learning. These are very different from programming."

MEETING THE DEMAND

Horswill and Sood have found a way to manage the escalating number of students who want to learn. In past quarters, students met in discussion groups led by a teaching assistant, but those sections were still overwhelmed with 80 students.

CODE NOT INCLUDED

Professor Jason Hartline wants his students to know that computer science is more than programming. Here are some other topics within the realm that Hartline teaches in CS 101:

- THEORY OF COMPUTATION
  what computers can compute
- ALGORITHMS FOR EFFICIENT COMPUTATION
  what to tell a computer to compute
- PROGRAMMING LANGUAGES
  how to tell a computer what to compute
- ARTIFICIAL INTELLIGENCE
  how a computer can perform like a brain
- COMPUTATIONAL BIOLOGY
  how DNA-based biological systems are like programs
- COMPUTER SYSTEMS
  how computers work together in networks, such as the Internet
- COMPUTER VISION AND GRAPHICS
  how computers gather and convey useful visual information
- HUMAN-COMPUTER INTERACTION
  natural ways to get what is needed from a computer

Thanks to funding from Charlene and Bob Shaw ('70), the department was able to hire 18 peer mentors to lead small tutorials with just six students each. The department also introduced an advanced section to challenge students who enter the class with programming experience.

An introductory course to computer programming, CS 111 is required for all computer science majors, but half of the class comprises non-majors—a proportion that grows every quarter. This winter quarter 2016, the class reached nearly 350 students—another number that steadily continues to grow.

Some take both courses to explore the world of computer science, dipping a toe into the vast field to decide if it's something they want to pursue. Others are simply interested in learning more about topics, such as algorithms, that are seeping into our everyday lives and becoming prominent in the daily lexicon.

"More and more, computer science is becoming something that everyone should know to be a functioning person in society," Hartline says. "You should understand that you are regularly interacting with learning algorithms. And you should know what that means."

Sood agrees. "Programming has moved into many fields beyond computer science. It has become necessary to have at least a basic introduction."

AMANDA MORRIS