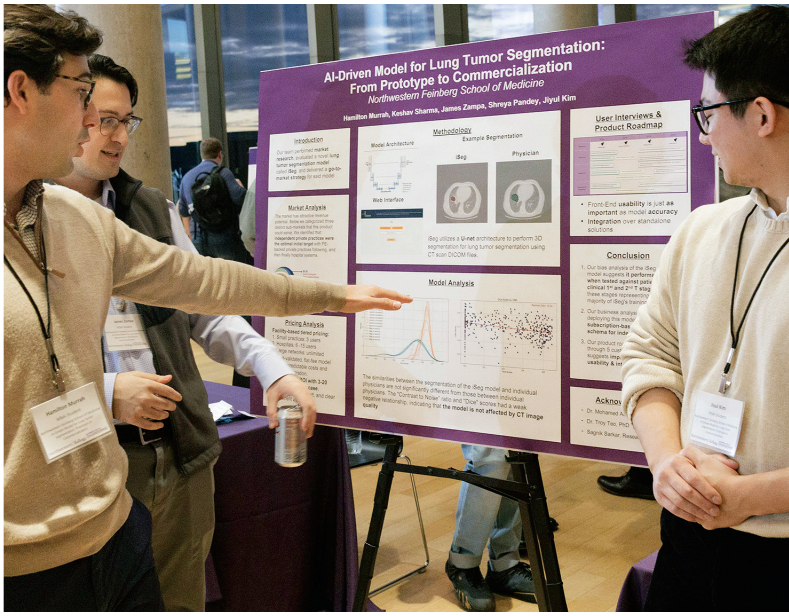
Cracking the Code: Preparing Students for an AI-Driven Future

Northwestern Computer Science faculty strike a balance between mastering core skills and leveraging generative AI in CS education.


AI-Driven Model for Lung Tumor Segmentation:
From Prototype to Commercialization
Northwestern Feinberg School of Medicine
Hamilton Murrah, Keshav Sharma, James Zampa, Shreya Pandey, Jiyul Kim
Introduction
Methodology
Market Analysis
Model Analysis
Pricing Analysis
User Interviews & Product Roadmap
Conclusion

NORTHWESTERN COMPUTER SCIENCE LEADERS SAMIR KHULLER AND SARA OWSLEY SOOD ARE NO STRANGERS TO TIDAL SHIFTS IN THEIR DISCIPLINE'S TECHNOLOGY.

In June 2007, Sood graduated from Northwestern Engineering with a PhD in computer science, advised by AI pioneer Kristian Hammond. The same month, Apple released the first iPhone—unlocking a new era of mobile-first computing and the global developer marketplace of the app economy.

A discipline marked by profound waves of innovation, computer science has always advanced at breathtaking speed. Few shifts have felt so sudden and consequential as the arrival of generative AI and its rapid integration into everyday tools. Systems like ChatGPT, GitHub Copilot, and Claude are transforming not only the tools of the trade, but also the very definition of programming. According to analysis by Deloitte, generative AI will likely be embedded into nearly every company's digital footprint by 2027.

The result is a field in flux—for students, researchers, and professionals alike. To excel in this rapidly changing landscape, students need AI literacy and readiness training.

"Just as we adapted to train students with the newest version control systems, integrated development environments, and tool kits, now we must ready our students to join the workforce fortified with extensive experience with generative AI tools, knowledge of current best practices, and a strong ethical framework on how and when they should be used in the software engineering process," says Sood, Chookaszian Family Teaching Professor of Instruction and associate chair for undergraduate education in the McCormick School of Engineering.

"JUST AS WE ADAPTED TO TRAIN STUDENTS WITH THE NEWEST VERSION CONTROL SYSTEMS, INTEGRATED DEVELOPMENT ENVIRONMENTS, AND TOOL KITS, NOW WE MUST READY OUR STUDENTS TO JOIN THE WORKFORCE FORTIFIED WITH EXTENSIVE EXPERIENCE WITH GENERATIVE AI TOOLS, KNOWLEDGE OF CURRENT BEST PRACTICES, AND A STRONG ETHICAL FRAMEWORK ON HOW AND WHEN THEY SHOULD BE USED IN THE SOFTWARE ENGINEERING PROCESS."

Sara Owsley Sood
Chookaszian Family Teaching Professor of Instruction,
Associate Chair for Undergraduate Education

Because the fundamentals remain key to building essential skills like algorithmic thinking, analytical reasoning, and navigating multiple layers of abstraction to define problems and design solutions, Northwestern CS students are generally prohibited from using generative AI tools in introductory (100- and 200-level) courses. In these courses—required for all CS majors and prerequisites for CS master's degree programs—students gain proficiency in multiple languages, tools, and programming paradigms. They also learn core concepts including debugging, time and space complexity analysis, abstraction, and writing secure and robust code.

Students in advanced (300-level and above) courses, however, have considerable latitude. In certain courses, faculty highly encourage or mandate the use of generative AI as long as it does not distract or deviate from the learning objectives. Each instructor determines whether and how to incorporate these tools and specifies the usage policy in the syllabus.

"The way we learn to think like computer scientists is in the practice of decomposing problems, designing programs, weighing efficiency tradeoffs, and struggling with bugs. If students outsource this process to AI, they will not master the fundamentals."

Samir Khuller Peter and Adrienne Barris Chair of Computer Science

Sood, Khuller, and the Northwestern CS faculty agree that striking the right balance and timing on the use of generative AI in the learning process is essential. At the heart of the department's teaching philosophy is the belief that education is a journey and shortcuts shortchange the learning process. Mastery requires more than getting the right answer or getting something done. By focusing on teaching how to think and reason, Northwestern CS equips students with skills that extend well beyond the classroom.

"Unfortunately, having generative AI tools at one's fingertips has great potential to deny students critical learning opportunities," says Khuller, Peter and Adrienne Barris Chair of Computer Science. "The way we learn to think like computer scientists is in the practice of decomposing problems, designing programs, weighing efficiency tradeoffs, and struggling with bugs. If students outsource this process to AI, they will not master the fundamentals."

In advanced courses, students also gain technical depth and explore areas such as human–computer interaction, quantum computing, computer vision, and security. They learn to approach challenges from both algorithmic and systems perspectives while exploring the evolving layers of the software and hardware stack. By rapidly mastering new problem domains, students take on increasingly complex work, innovate effectively, and strengthen their ability to navigate ambiguity and uncertainty.

"Generative AI tools have great potential to amplify students' learning in upper-level courses," Sood says. "From quickly prototyping a system to transforming data into a different format to producing graphs or visualizations of data, generative AI tools can alter what students can accomplish in project-based courses, enabling them to showcase their creativity and innovate."

Photos by Carlos Javier Ortiz and Joel Wintermantle

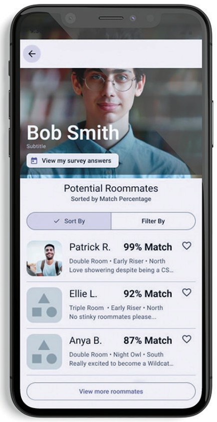
Inside the Classroom: Leveraging AI in Advanced Courses

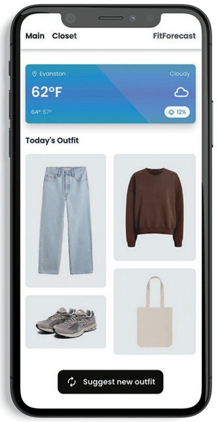
DEVELOPING SOFTWARE THROUGH REFLECTIVE ANALYSIS AND ITERATION

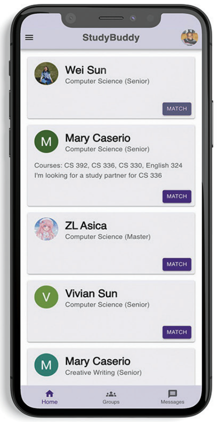
- **COMP_SCI 392: Rapid Prototyping for Software Innovation**
- **COMP_SCI 394: Agile Software Development**

INSTRUCTOR **Chris Riesbeck, Associate Professor of Computer Science and Director of the Master of Science in Computer Science Program**

Except for a few learning exercises, Chris Riesbeck doesn't restrict students' use of generative AI in his team-based, full-stack mobile and web software application project courses. "If it's permitted in industry, it's permitted in the class," he says.


Bob Smith
Subtitle
Potential Roommates
Sorted by Match Percentage
Patrick R. 99% Match
Double Room • Early Riser • North
Love showering despite being a CS...
Ellie L. 92% Match
Triple Room • Early Riser • North
No stinky roommates please...
Anya B. 87% Match
Double Room • Night Owl • South
Really excited to become a Wildcat...




Wei Sun
Computer Science (Senior)
Mary Caserio
Computer Science (Senior)
Courses: CS 392, CS 336, CS 330, English 324
I'm looking for a study partner for CS 336
ZL Asica
Computer Science (Master)
Vivian Sun
Computer Science (Senior)
Mary Caserio
Creative Writing (Senior)

Third- and fourth-year undergraduates and master's degree students come to CS 392 and 394 with strong programming backgrounds. Coding takes a backseat to applying critical reasoning, reflective analysis, and cross-functional team-building skills to iteratively develop prototypes that meet client requirements. Mid-class surveys show that most students use generative AI tools, but sparingly. So far, students who learned to code traditionally find AI-powered coding tools disruptive to their flow.

"Students want to solve the program logic themselves," Riesbeck says. "But they use generative AI to help explain bugs, fix errors, and handle messy boilerplate code like CSS."

Riesbeck encourages experimentation with safe, business-oriented uses of generative AI, such as writing module tests, mocking up demos, and comparing manually written code with AI outputs. This fall, CS 392 students completed one project using generative AI extensively to create several prototypes rapidly. "Generative AI is great for building throw-away prototypes and iterating on ideas," Riesbeck says.

EXPERIMENTING WITH AI-ASSISTED CODING

- **COMP_SCI 397: Applied AI for Software Development**

INSTRUCTOR **Hamilton Murrah (MBAi '25), Adjunct Professor of Computer Science**

In this course, a mix of fourth-year undergraduate, combined BS/MS, and master's degree students draw on their foundational programming knowledge to experiment with generative AI using increasingly sophisticated AI-assisted coding tools. Student teams first test how tools like ChatGPT, Cursor, and Claude handle tasks such as debugging, refactoring, and exploring natural language coding. They then pair-program a full-stack, Twitter-style application, documenting each step of their decision-making processes via an architecture design document.

"You have to use the right tool for the right job," Hamilton Murrah says. "What's really important is having people who know how to interact with and get the best results from generative AI systems while also being able to think critically about what is needed to solve a specific problem."

ANALYZING AND EXPLAINING AI MODEL BEHAVIOR

- **COMP_SCI 396: Fairness in Machine Learning**

INSTRUCTOR **Zach Wood-Doughty, Assistant Professor of Instruction**

No single metric of fairness applies to all machine learning (ML) models. That makes reasoning abstractly about algorithmic bias and its high-stakes implications difficult. In the Fairness in Machine Learning course, student teams design, evaluate, and critique ML autograders using a published dataset of anonymized essays. As a class, they set goals and requirements of the AI-assisted graders, which leverage large language models to assess writing assignments.

To simulate fairness challenges, Zach Wood-Doughty randomly introduces modifications—such as sophisticated vocabulary, misspellings, or grammatical quirks—that could create irrelevant or unintended shortcuts for grade predictions. Students then apply interpretability methods to uncover and explain the model's behavior, exploring how altering the model, data, or optimization affects fairness.

"If we artificially inflate grades for essays with four-syllable words, we're building in bias that equates fancy words with quality," Wood-Doughty says. "The model becomes a sandbox for how we detect this bias and what the implications are for other data categories. Suppose you're deploying this model to score SAT essays. What measures would you prioritize, and what tradeoffs in fairness, accuracy, and performance would you accept?"

ITERATING TO MAXIMIZE USER VALUE

COMP_SCI 397: Software Studio

INSTRUCTOR **Anastasia Kurdia, Professor of Instruction and Assistant Chair of Computer Science**

In this course, Anastasia Kurdia welcomes all modern tools. Students work in agile development teams to iteratively build a public-facing software-as-a-service application and use GitHub Copilot, a context-aware code generator, to enhance productivity. Kurdia intends the course to reflect the methodology, programming languages, and development tools in software development jobs—for example, test-driven development, collaborating via version control software, and pair programming.

"Students demonstrate technical skills, fluency, and fearlessness—when they have to deal with six technologies that are new to them, they're not afraid of the seventh," she says. "What we are able to build in one quarter used to take two semesters. Students build on each other's strengths and deliver working projects because they are not too busy debugging unfamiliar code."

Team members alternate roles from sprint to sprint, taking turns as a customer representative seeking to prioritize features. "Developers learn from users," Kurdia says. "Understanding how users interact with a product allows us to maximize value with each iteration. One can envision and build almost anything, but success comes from deploying, learning, building, and deploying again."

INTEGRATING TOOLS TO BUILD USER-CENTRIC SOLUTIONS

MSAI 495 SPECIAL TOPICS: AI Platforms

COORDINATOR **Mohammed Alam, Assistant Professor of Instruction and Deputy Director of the Master of Science in Artificial Intelligence (MSAI) Program**

INSTRUCTORS **William Johnson, Arvind Periyasamy, Julia Heseltine, and Pablo Salvador Lopez (Microsoft)**

Given the flexibility and scalability of industry-standard platforms like Amazon Web Services (AWS), Google Cloud, and Microsoft Azure, developers are increasingly building end-to-end applications in the cloud. "One way our students can differentiate themselves from the moment they join industry is by demonstrating fluency in how to develop user-centric, generative AI solutions in the cloud," Mohammed Alam says.

In the five-week course, MSAI students train with a Microsoft team to develop, deploy, and manage a chatbot using Azure's AI tech stack. Students select and integrate the relevant portfolio of services that fulfill the project scope requirements—including generative AI text generation, machine-learning image recognition, AI workflow automation, and dynamic problem-solving. They practice skills, including connecting a company's database with the Azure platform and building a retrieval-augmented generation architecture that trains the chatbot's large language model with domain-specific, proprietary enterprise data. "Students are connecting tools and APIs to produce one big solution for the problem they define," Alam says. "Platforms like AWS and Azure provide the blocks and MSAI students build powerful AI systems while developing a deep understanding of how those blocks fit together."

TACKLING LIVE BUSINESS PROBLEMS CROSS-FUNCTIONALLY

MBAI/MSAI-490: Capstone Project

INSTRUCTORS **Mohammed Alam and Andy Fano, Clinical Professor of Computer Science and Director of the MBAi Program, a joint degree from Northwestern Engineering and Northwestern's Kellogg School of Management**

Hybrid teams of MBAi and MSAI students partner with industry clients in this quarter-long capstone focused on finding scalable solutions to complex business problems, often through generative AI and large language models. But Andy Fano cautions against putting the cart before the horse. "It's not about plugging in a technology, it's about solving a live, domain-specific problem," he says. "Teams sharpen underspecified objectives, negotiate project scope, analyze workflows, manage noisy or incomplete data, and articulate the business value of a technology solution."

This fall, 17 teams collaborated with 17 companies, from Fortune 100s to startups. Baxter and Ecolab are exploring new ways to train customers on their devices and systems, while CDW is pursuing strategies to ensure product page accuracy.

Past projects tackled challenges such as structuring hospital pharmacy data for Accenture, designing sugar-harvesting automation for John Deere, and developing an AI textbook generator for ViewSonic. Fano emphasizes that while some graduates will build generative AI tools with major tech firms, most will join companies that use an evolving range of AI and ML capabilities. "Generative AI is the big new thing, but in terms of what's actually deployed in the business world, the value is in the more traditional machine learning that we take for granted, like credit card fraud detection. Things might change."

MICHELLE MOHNEY