

Healthcare Data Sharing (*continued*)

for data to conform to these standards, ultimately focusing on supporting patient treatment. Tammy Toscos, situated between the realms of Margarita and John, discussed her lab's work with public health data and qualitative data integration. While utilizing electronic health record data for research and support systems, her team emphasizes social science and human-centered design in their core research areas.

As we navigate the intricate landscape of healthcare data sharing, the insights from this CRA-I roundtable can serve to guide stakeholders from academia, industry, government, and medical practice towards collaborative solutions. The challenges are acknowledged, but so too are the transformative possibilities that lie ahead in reshaping the future of patient care, research, and policy-making. It is clear that there is more to discuss as a community in this space. CRA-I will be hosting an in-person workshop in this area in 2024. Please stay tuned and **reach out if you have any questions, comments, or interest in participating in the in-person workshop.**

Expanding Career Pipelines by Unhiding the Hidden Curriculum of University Computing Majors



by Philip Guo, Associate Professor of Cognitive Science, UC San Diego

Everyone who attends college learns a formal curriculum by taking classes, but there is also a *hidden curriculum* just beneath the surface of their college experience. This hidden curriculum consists of the “unspoken lessons, norms, values, and perspectives that impact learning and academic performance. [It is] often implied and not explicitly taught, which poses various accessibility and equity barriers.” [1] As an example, consider Alicia, a student from a low-income family who is the first in her family to attend college. Even though she earns good grades in her classes, she notices that her classmates from more well-resourced backgrounds seem to have a much easier time getting ahead in their careers. They appear to have access to some secret “insider knowledge” and are somehow able to find more opportunities, network more fluently, obtain prestigious internships, and ultimately get good jobs right after graduation. In contrast, she does not know how to even start approaching senior students, professors, and alumni to discover such hidden opportunities. There are professionally-oriented clubs and student mentoring organizations on campus, but she lacks the self-confidence to join them since the students there seem too experienced and intimidating. Alicia feels frustrated since she excels in the formal curriculum by earning good grades in her classes, yet she struggles to find professional and career opportunities because she doesn't know how to access the hidden curriculum that many of her peers seem to have learned *outside* of classes.

There are millions of students just like Alicia attending universities around the world. For instance, amongst our 33,000 undergraduates here at UC San Diego, 24% are from underrepresented ethnic groups, 38% are first-generation college students, and 33% transferred in from community colleges. Research shows that these students are more likely to face challenges in unlocking career opportunities than classmates who come from more privileged backgrounds, due in part to them not knowing as much about the hidden curriculum. This harms computing career pipelines because these students may not have as equitable access to the most desirable jobs after graduation. Thus, over the past five years my students and I have developed a two-pronged approach to addressing this inequity: 1) creating a student-led mentoring guide to uncover aspects of the hidden curriculum, 2) explicitly teaching other aspects in formal computing courses. This article summarizes both of these initiatives.



Unhiding the Hidden Curriculum (*continued*)

A Student-Led Guide to the Hidden Curriculum of Computing Majors

In an ongoing project created by an undergraduate student, [Kendall Nakai](#), we took a unique student-centered approach to uncovering the hidden curriculum of computing-related majors to help students make the most of their college experience and position themselves better for the job market. We noticed that existing mentoring resources are often created by university staff such as department advisors or career center employees, but these official school resources lacked a lot of the subtle unwritten insights that students needed. Kendall's intuition was that a student-created guide could supplement these official resources and provide information that they lacked. To test this intuition, she interviewed dozens of students and recent alumni to compile their collective wisdom into a 50-page online guide to the hidden curriculum of our HCI (Human-Computer Interaction) and Design specialization at UC San Diego.

This guide contains advice on how to navigate courses, how to talk to professors, how to approach peers for help, how to create a project portfolio to prepare for jobs, how to look for internships, and eventually how to get full-time jobs in the technology industry. To give a sense of what it covers, here is an excerpt from its table of contents:

What are Hackathons and Designathons?	25
I have to apply to attend a hackathon... am I qualified?	26
How do I create or redesign features of apps or websites I use?	26
How do I conduct a personal research project?	27
■ How do I find an internship?	28
I'm not a junior yet, so why should I apply for internships?	28
Okay I'm interested in applying, what should I look at?	28
When do I apply?	29
I'm a junior and I'm looking for an internship or a senior looking for a job...what should I do?	29
Every internship or job description says that I need to have skills I haven't learned yet...can I still apply?	29
I've met people that have worked at big "brand name" companies...should I try to work there too?	29
Recruiters might not know that Cognitive Science as a major applies to the role...what should I put?	30
I can't seem to get any internship interviews...how do I get my application in front of a person?	30
How do I talk to recruiters at career fairs?	31
The chat isn't going so well...the recruiter doesn't seem interested in me or says I'm not eligible for positions...what do I do?	31
How do I remember all the people I've met?	32
Career fairs seem big and scary. Got any tips?	32
I'm not feeling motivated...How do I deal with "ghosting" or rejection?	33



Unhiding the Hidden Curriculum (*continued*)

Note how this guide adopts a casual conversational tone as if a peer mentor is talking directly to the reader, in contrast to university resources that are written with an ‘official-sounding’ authoritative tone. Zooming into individual sections, the guide provides specific guidance for common scenarios that computing students encounter. For instance:

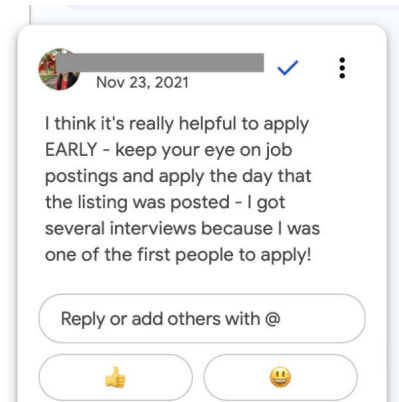
3. **APPLY** APPLY APPLY – to jobs, internships, part-time or contract positions

Every internship or job description says that I need to have skills I haven’t learned yet...can I still apply?

YES! The skills listed are always good **guidelines** to follow or work toward but you don’t need to meet every single one.

Usually knowing how to use at least one wireframing/prototyping tool (like Figma, Adobe XD, and Sketch) and having basic web development (HTML, CSS, JavaScript) knowledge will help you. The rest you can somewhat pick up on the job.

💡 If you have concerns, [ask the people you talk to](#) advice on what they think of your current skills, resume, portfolio and see what advice they have to advance your skills to what job apps look for.



And since this guide is simply a web-based Google Doc, other students can chime in with comments to help their classmates, as shown in the above screenshot. From time to time, Kendall curates the most insightful comments and integrates them into the text itself.

This guide also provides step-by-step scripts for helping students navigate unfamiliar settings, such as when they first join a research group:

I just started research with a professor, how can I stand out?

Advice	Example
Communicate before being asked to (take initiative)	If every week before the lab meeting, send a summary to the professor or graduate student “I’d like to talk about _____, I made progress in _____, and here are some questions I have about _____.” This alone will help you stand out a lot!
Show progress	“I completed X task and started Y task this week. Here is my progress. Before I continue with Y task, I have some clarifying questions...”
Be realistic about the amount of time you can commit	“I can commit 8 hours a week – 2 hours a day 4 days a week except maybe less during midterms. Is this okay?”

Note that students from certain backgrounds (e.g., those with parents who have graduate-level degrees) may already know how to do these things since they grew up with early exposure to these types of professional interactions. However, many college students, especially those from underrepresented groups, did not grow up with this kind of hidden curriculum knowledge such as how to approach authority figures like professors. This guide surfaces such knowledge to help these students overcome some of the psychological barriers to getting started.

Since its release in Fall 2021, this guide has been viewed over 2,700 times by students here at UC San Diego, and professors have also started sharing it in some of their course syllabuses. In addition, students have been passing around links to it online so that it



Unhiding the Hidden Curriculum (*continued*)

has spread via word-of-mouth. Some have personally messaged Kendall to thank her for providing this resource that is helpful even when they cannot find an in-person mentor. Readers have said:

"I honestly wish I had come across a resource like this early on in my college career. The 'How do I find an internship' section really hit on a lot of things I had to figure out myself such as when to start applying. Reading through this guide made me feel better about not knowing how the internship application cycle worked until my junior year because it really is something that isn't explicitly taught, but rather something you learn from your parents or from experience."

"I have learned how to connect and talk to the people in my field. In fact, after reading the guide, I messaged many people via LinkedIn and had a call with a current Google software engineer to learn about what he does and what I can do to be prepared."

"I honestly wish I had come across a resource like this early on in my college career. The 'How do I find an internship' section really hit on a lot of things I had to figure out myself such as when to start applying. Reading through this guide made me feel better about not knowing how the internship application cycle worked until my junior year because it really is something that isn't explicitly taught, but rather something you learn from your parents or from experience."

From this project we learned that it is critical for mentoring resources to be written in an authentic student voice in order to make them the most relatable. Our readers appreciated how this guide was written by a peer and not an authority figure, with one responding: *"Reading this guide made me tear up a little. I love the amount of passion, pride, and care that went into making this guide. I really appreciate that and it's breathtaking to see the amount of work that was put into this guide for the benefit of future students."*

Related to above, we also believe that students relate best to a guide written by their own classmates, since some aspects of the hidden curriculum may be specific to a school or even within a particular department. Thus, we envision a future where students at each school write their own hyperlocal guides with content and style that resonate most with their classmates. For more details about the design process and evaluation of this project, [check out our ICER 2023 research paper \[2\]](#).

Explicitly Teaching the Hidden Curriculum in Computing Courses

In parallel with Kendall's undergraduate-led guide, my graduate students and I have also been developing a series of HCI (Human-Computer Interaction) courses that teach not only technical content but also aspects of the hidden curriculum in more detail than what the guide introduces. We embarked on this curriculum design journey five years ago since we noticed that while there are many HCI courses that teach theoretical principles and how they apply in practice via class projects, many students were not able to make the leap from finishing these classes to using their projects as the basis for finding jobs. These classes take students most of the way, but there is still a *last-mile problem*.

The last-mile problem is a term from communications, infrastructure, and logistics design that conveys the difficulty of fully connecting services end-to-end. For instance, a subway station may be located over a mile away from many commuters' homes or offices, so they also need to bring bicycles or skateboards with them to cover that last mile. In the context of project-based computing courses, the last-mile refers to the gap between a finished class project and being able to use it effectively to look for jobs. Many project-based courses provide the raw materials for students to present in their portfolio when job-hunting, but they never teach students how to write up their work in a format that appeals to modern-day employers. Instead, students are left to figure this out by themselves. As a result, even though they have put in all the work to complete a project, many find it hard to



Unhiding the Hidden Curriculum (*continued*)

traverse that vital “last mile” to turn it into a compelling addition to their portfolio that employers value. This gap represents a kind of hidden curriculum since nobody explicitly teaches students how to leverage their class projects to look for jobs.

In my HCI courses, I address this last-mile problem by teaching students how to write up a **design case study** and upload it to a real working portfolio website as their end-of-term deliverable. A case study is a blog-post-like webpage that shows how a student adopted a user-centered design process from the start of a project all the way to the finished product; it explains the analytical reasoning behind all the various design decisions made along the way. Employers have told us that these case studies are the most important part of a student’s portfolio when they apply to tech jobs such as UI/UX Designer, Product Designer, or UX Researcher. In addition, I employ a team of undergraduate TAs who provide mentorship by critiquing students’ portfolios and resumes, offering insider advice on how to look for internships and jobs, and being there for emotional support and reassurance. Taken together, our pedagogical goal is to surface aspects of the hidden curriculum via both explicit instruction and undergraduate-led peer-to-peer mentorship.

Over the past five years, my graduate students and I have been teaching two large undergraduate courses that embody this philosophy: 1) *Data-driven UX/Product Design* is a course where students work on a team design project that culminates in a new case study that every student writes up on their own portfolio website. That way, as soon as they finish the course they are ready to present that portfolio to employers. My student Sean Kross (who co-developed then taught this class with me) and I wrote up a [SIGCSE 2022 paper about this course’s pedagogical philosophy](#) [3]. 2) *HCI Portfolio Design Studio* is a course where students get a chance to take their completed projects from prior courses they have taken and write up their results as case studies to put on their portfolio website. In addition, students practice making slide deck presentations for design-related job interviews, refining their LinkedIn profiles, and learning other career development skills. The goal of this course is to bridge that last mile between projects that students completed in prior courses and what is needed to apply for jobs.

Course review comments from students have been encouraging, with some even indicating that they were able to directly use what they produced in these classes to get their first internship or job. Here are a few representative comments:

“Dear Future Students, TAKE THIS CLASS! It is hands-down the most practical class I have taken in my three years at [my university]. This is the only class I’ve taken so far in my three years that might actually help me find a job as it guides you through the process of writing a case study for your portfolio.”

“This class [...] stood out to me for how practical the lectures were. In other classes here, I haven’t had any professors telling me what the trajectory of an aspiring [HCI] student could be like in addition to going through the different potential paths we could take. I also appreciated the fact that the entire term led up to us creating a case study that would be genuinely useful in our future job-search process.”

Lastly, each year I teach a smaller seminar course called *HCI Technical Systems Research*, which introduces undergraduate students to PhD-level research in the field of HCI. Aside from discussing the technical content of papers, we spend an equal amount of time talking about the behind-the-scenes of how the professional research enterprise works. This includes topics such as grant funding, the peer review publication system, how academic careers are structured (e.g., Ph.D. programs, postdocs, tenure-track professorships), industry versus academic research environments in computing, technology transfer of research into products, and other domain-specific insider information that undergraduates usually do not get access to. Similar to my other classes, my goal is to reveal aspects of the hidden curriculum that students would normally not learn in classes. In this case, I hope to better prepare students to join research labs and apply for Ph.D. programs in the future.



Unhiding the Hidden Curriculum (*continued*)

Note that much of what I teach in these courses is not technical content but is rather implicit ‘behind-the-scenes’ knowledge. My motivation for introducing this kind of hidden curriculum in formal courses is that it can potentially be the most equitable, since all students can access these courses if they meet the prerequisites. In contrast, if this were not taught in the classroom, then certain groups of students (e.g., those who are well-resourced or better-connected on campus) would disproportionately get more access to such information via their peer groups.

Parting Thoughts

Both Kendall’s student-created guide and my HCI courses aim to level the playing field by sharing the sorts of unspoken knowledge that well-resourced students currently have better access to. Some of the career development advice we present may seem “obvious” and “simple” to those who grew up with parents, friends, and older mentors who taught these things to them. But for many students, especially those from underrepresented groups, these insights may not be obvious so that is why it is important to teach them in systematic ways. Looking forward, I believe it is our responsibility as educators to continue finding ways to reveal more aspects of the hidden curriculum to our students.

Acknowledgments

Thanks to Kendall Nakai for leading the hidden curriculum guide project and to Sean Kross, Ian Drosos, and Sam Lau for helping me develop and teach the HCI courses described in this article. This material is partly based upon work supported by the National Science Foundation under Grant No. NSF IIS-1845900 and by the Google exploreCSR program.

About the Author

Philip Guo is an associate professor of Cognitive Science and (by affiliation) Computer Science & Engineering at UC San Diego. His research spans human-computer interaction, data science, programming tools, and online learning. He studies how people learn computer programming and data science, and he builds tools to help people better understand code and data.

References

- [1] Nicole Campbell et al. 2022. Uncovering the Hidden Curriculum. <https://hiddencurriculum.ca/>
- [2] Kendall Nakai and Philip J. Guo. Uncovering the Hidden Curriculum of University Computing Majors via Undergraduate-Written Mentoring Guides: A Learner-Centered Design Workflow. ACM Conference on International Computing Education Research (ICER), 2023. [PDF]
- [3] Sean Kross and Philip J. Guo. Five Pedagogical Principles of a User-Centered Design Course that Prepares Computing Undergraduates for Industry Jobs. ACM Technical Symposium on Computer Science Education (SIGCSE), 2022. [PDF]