CS FOR ALL ACADEMIC IDENTITIES [*]

Madeline Zug, Hanna Hoffman, Forest Kobayashi, Miles President, Zachary Dodds
Computer Science Departments, Pomona College and Harvey Mudd College
Claremont, CA 91711
madelinezug@gmail.com, {hhoffman, fkobayashi, mpresident, zdodds}@hmc.edu

ABSTRACT

"CS for All" has set computing on an unusual journey. Those words ask CS to change: to grow from a compelling discipline and useful mindset into a full-fledged human literacy. Just as cogent writing, critical reading, and compelling speaking are today's hallmarks of literacy, so too will leveraging computing for insight become part of the goals and expectations we all share. This paper considers how Computer Science, both as a discipline and as an academic department, can support this journey. To map the landscape, we first survey the extent of computing's current curricular reach – beyond CS departments – at a sample of fifty U.S. institutions. We then present findings from three experiments, local to our institutions, which explored interdisciplinary course structures. Both the local and the global overviews suggest that CS departments have, now, a unique opportunity to help smooth computing's transformation into a modern literacy. It's in the best interests of all disciplines, together, to bring computing, its resources, and its roles into their distinctive identities.

MOTIVATION

"CS for All" is a compelling call to action, but "for All" can be read in at least two ways. On one hand, it captures our goals for universal inclusion: everyone is capable of powerful, purposeful self-expression via computing's toolsets. On the other hand, the phrase can raise alarms: like it or not, here comes computing. In 2001, the New York Times published an article titled "All Science is Computer Science," and that sentiment that has reverberated since. [6]. The words suggested a possibly overzealous "here to

---

help" attitude among CSers, though not always certain of the form that help might take — or how helpful it would really be. XKCD's take is apt [8].

The words "CS for All" can reinforce a perception that CS is encroaching on other disciplines, intentionally or not. Put another way, computing's utility can seem like a threat to some forms of *academic identity*. Here, "academic identity" is a term that refers to one's self perception as a learner: what fields are of interest, what efforts demonstrate that interest, and what ambitions elicit those efforts [10]. [1] This is something that we, as a field, must be cognizant of — for in an educational setting, people holding non-CS identities are precisely those who determine computing's future reach. Computing will thrive to the extent that its mindset and toolset become part of other disciplines' academic identities. Today's CS community neither owns nor controls computing's future – but we do have a unique opportunity to help guide its growth and reach. 1 "Major" or "department" capture only the administrative senses of academic identity.

In this light, "All Science is Computer Science," is meant in the same sense that "All Science is cogent writing, critical reading, and compelling presentation." After all, no matter what discipline one chooses to study, practitioners rely on verbal literacy. All academic fields embrace these skills, weaving them through their students' experiences in ways that amplify their disciplines' goals and identities. We believe computing, i.e., the creation of task-motivated computational artifacts, is now joining this group of literacies, both in academic and professional contexts.

THE CURRENT NATIONAL LANDSCAPE

Methods

In our investigation, we examined the currently-published curricula from Table 1's sample of 50 institutions. We used the method of analysis described by Bowlick et al. to examine the ways in which different schools interface with computation on a course level [5]. In a nutshell, this entailed searching departments' curricula for programs and courses containing interdisciplinary content, as well as non-CS programs with computing content. Table 2 shows how the courses/programs examined were placed into one of five categories: (1) CS+X double majors or CS minors, (2) CS+X courses, (3) Interdisciplinary majors (4) Flavored CS1 courses, and (5) CS2 for non-CS majors.

[1] "Major" or "department" capture only the administrative senses of academic identity.

Table 1: Institutions examined as to CS curricular offerings across academic identities

Schools examined									
MIT	Stanford	CMU	Harvard	Caltech	UPenn	Yale	GA Tech	Rice	Brown
Berkeley	Duke	UM Ann Arbor	Columbia	Dartmouth	UCLA	USC	Cornell	Carleton	Vanderbilt
UI U-C	Wash U	Harvey Mudd	Bowdoin	WPI	Swarthmore	Pomona	Tufts	RPI	UT Austin
U Chicago	Amherst	Northwestern	UVA	Williams	Notre Dame	Haverford	Northeastern	VA Tech	SIT
UCSB	Emory	RHIT	Texas A&M	John's Hopkins	U Wisconsin	UW	SMU	Case Western	UCSD

Table 2: Frequency of five disciplinary overlaps

Category	# Schools	Primary Agent
CS+X double majors	50/50	Student
CS+X courses	50/50	Faculty
Interdisciplinary majors	27/50	Depts.
Flavored CS1	21/50	Depts. +
CS2 for non-CS majors	8/50	Depts. +

Table 3: Outcomes by CS1 Flavor: courses and major yield

CS1 Flavor	# Courses Taken†		% Majoring in	
	CS	Bio	CS	Bio
Unflavored	4.0	1.8	17.2	3.1
Bio-Flavored	4.3	3.8	15.0	18.7

† This is an average, and includes the CS1 course taken.

Findings

CS+X double majors and CS minors. All fifty of the schools we surveyed had options for students to either minor in CS or double major in CS and a non-CS discipline in parallel, without having the connections between the disciplines explicitly established by coursework.

CS+X courses. All fifty schools also offered at least one CS+X course. These efforts require more than student initiative: typically, faculty from multiple departments must collaborate to develop curriculum in unfamiliar territory [7]. We found that Biology was the most common "X," with over 40 of the schools offering some kind of computational biology course.

Interdisciplinary majors. Slightly more than half of the schools offered interdisciplinary majors. Such programs are distinct from CS+X in the way they approach the intersection of the two disciplines. Interdisciplinary programs explicitly explore the connection between CS and X in an "integrative experience" — a project, thesis, or course that requires students to apply CS to their X. Implementing such a program usually requires active collaboration on a departmental level. Because individual faculty members typically have more flexibility than entire departments, it's not surprising that interdisciplinary programs are less common than CS+X courses.

Flavored CS1. Flavored CS1 comes in two main forms: upper-division and lower-division. Upper-division CS1 courses cater to upperclassmen in a non-CS discipline where CS skills are desired. These students have already developed a non- CS identity and substantive knowledge in their domain, so they are able to apply CS1 skills to solve discipline-specific problems that underclassmen don't have the experience to address. Example of upper-division flavored CS1 courses include CS+English, computational thinking for architects, and music and computation [4, 9, 12].

Lower-division flavored CS1 courses aim to engage more students in computer science before they are tend to have a strong academic identity. Three of the many examples of lower-division flavored CS1s include six distinct contexts offered at Union College, a socio-techno CS1 at Williams, and a computer music course at Yale [3]. Flavors we found most often through our analysis include biology, game development, CS for engineers (usually taught in Matlab), and CS for scientists (most often taught in

Python or Matlab). Fewer than half of the schools surveyed offer lower-division flavored CS1 within the CS department.

CS2 for non-CS majors.

Fewer than ten of the schools surveyed offered a CS2 course for non-majors (CS2-NM). A common theme of the courses we found was application of CS — data visualization, web scraping, and directory navigation. Popular languages (often used in combination) included Python and R.We suspect that such courses are relatively rare not only because they entail shared ownership of computing, as with flavored CS1, but also because they explicitly support students with non-CS academic identities — curricular areas where specialty matures into literacy.

LOCAL EXPERIMENTS

We have experimented at our institution with the above interdisciplinary opportunities. We discuss here the effectiveness of flavored CS1 and CS2-NM, and we describe bridges and injections created to meet demand for computing skills. Bridges are CS1 extra-credit assignments introducing computing concepts not covered in CS1; injections are collaboratively-created curricular facets used in non- CS courses.

A biologically-flavored CS1

Since fall 2009 we have offered a biology-flavored CS1 in addition to the standard CS1 course. Students with no previous CS experience could choose between "unflavored" and biology-flavored CS1. Over the last eight years, 839 students took unflavored CS1 and 193 took biology-flavored CS1. For each student, we gathered data on their CS1 course, the CS and biology classes they took in subsequent semesters, and the grades they earned in those classes.

We first examined whether taking biology-flavored CS1 was a detriment for those who continued studying CS. We found that students continued to take CS courses and declare CS majors at similar rates, regardless of which CS1 course they took, as shown in Table 3. Also, students who took the biology-flavored CS1 fared just as well as their peers who took unflavored CS1 in subsequent CS classes, as Table 4 displays. This suggests that our biology-flavored CS1 prepares its students for later CS courses at least as well as our unflavored CS1.

Next, we wanted to see whether biology-flavored CS1 was allowing students to retain and foster their interests in biology. Table 3 shows that students who took unflavored CS1 took an average of 1.8 biology courses and declared biology majors at a rate of 3 percent. Students who took biology-flavored CS1 took 3.8 biology courses and declared biology majors at a rate of almost 19 percent. This suggests that students who arrived with biology interests retained this identity through biology-flavored CS1. Taken together, these data suggest that flavored CS1 is an effective way to let students retain and/or develop a non-CS academic identity, while still teaching the skills needed to succeed in later CS courses.

Although flavored CS1 is a good way to give students with non-CS identities an introduction to computing concepts, it often doesn't provide as much expertise as students

would like in order to apply computing in their non-CS academic work. To address this concern, we explored ways of teaching computational skills to non-CS majors beyond CS1: (1) bridges, which are built out of CS1 in the general direction of other disciplines, (2) injections, which are inserted into non-CS courses, and (3) CS2 for non-majors, which is a CS course aimed at non-CS majors with a CS1 prerequisite. The bridges and injections described below are freely available [1].

Table 4: Performance in Future CS courses

CS1 Flavor	Percent Taking		Avg. Grade†	
	CS2	CS3	CS2	CS3
Unflavored	69.5	40.0	3.30	2.83
Bio-flavored	71.5	47.2	3.30	2.84

† Grades given on the four-point scale.

Table 5: New Bridges Built Out of CS1

Subject	Assignment
Mobile Development	Java, Android Development, Swift, iOS
Web Development	Ruby, HTML/CSS
Engineering	Matlab, Arduino
Math/Statistics	Euler's method animation engine, LaTeX, R
Art	Processing, Arduino meets Processing

Bridges

One of the most common complaints about our CS1 course is that it is taught in only one language, Python; students want exposure to more languages. Bridges are one to two hour extra credit assignments offered to students in CS1 with the goal of providing an introduction to new languages and skills that could be more directly applicable to their academic identity than those learned in CS1. Some of the bridges built and offered for these purposes are shown in Table 5.

Injections

In pursuing this work, we interviewed many professors outside of computing about their relationship to that field. Several sought curricular injections that could be used in their courses. Injections are assignments or demonstrations with a computational component that are integrated into a non-CS course [3]. We developed these collaboratively, creating assignments and demonstrations in the context of each course's and discipline's independent curriculum. The injection topics included 3D mechanics using GlowScript, an n-body simulator using the Barnes-Hut algorithm, history-of-science searching via the NYTimes archives' API, an R tutorial, and a linear-algebra animation engine.

CS2 for NonCS Majors

In an effort to understand the types of non-CS students who are interested in CS2, we analyzed student major data from the second course in our institution's intro sequence from 2009 to 2017. Engineering, CS, and CS-Math majors account for 51% of the total enrollment of 1643 students, non-CS majors account for 38%, and undecided students 11%.4 The results for the major breakdown of non-CS students are shown in Figure 1. In the early 2010s, CS2 demand from non-CS majors was dominated by students with a STEM identity. More recently, demand for CS2 has increased among economics, politics, and other social science majors.

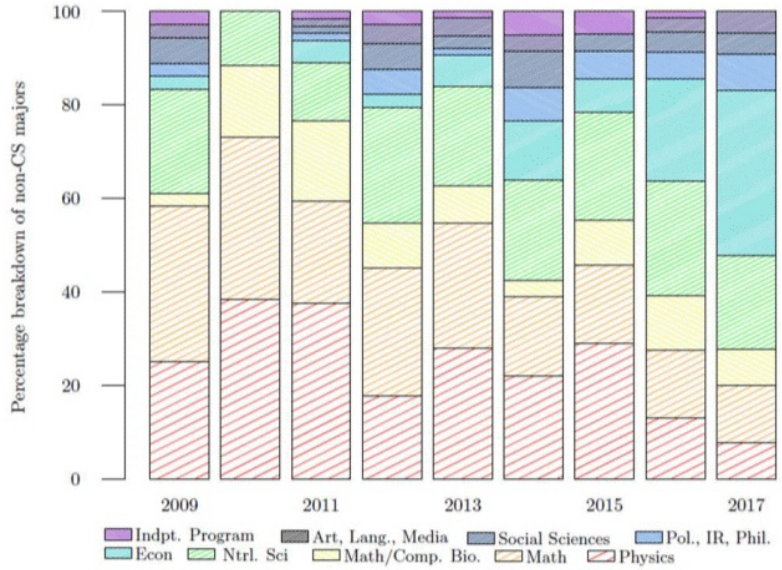

Figure 1. Changing distribution of interest in CS2, 2009-17

Figure 1 clarifies the trend from red-and- orange hashing (fields traditionally considered to have a computational overlap) to green-and- blue fill (fields more recently embracing computing as part of their undergraduate experience) is clear. Specifically, the majors represented each year, listed from bottom to top are physics; mathematics; computational or mathematical biology; other natural sciences; economics; politics, international relations, and philosophy; social sciences; art, languages, and media studies; independently-created programs of studies. Taken as a whole, our institution sees clear evidence that the demand for computational skills, well-established to be growing, comes from students with increasingly diverse academic identities.

To serve this growing population, our institution began offering a CS2 course for non-CS majors in 2015. A syllabus excerpt reads: "it is a course that builds computing skillsets and mindsets for all students wanting to leverage computing's resources for deeper insights into a their own (non-CS) field of interest." Student surveys taken corroborate this evidence of increasing demand from non-CS fields, reinforcing that computational skills are becoming one of the foundational literacies for more and more academic identities. For example, from a media studies major: "I think CS is extremely relevant to my major and would love for the media studies students to feel less intimidated by CS" or, from a politics major: "I see CS as incredibly relevant to data analysis and data visualizations. Politics is a series of narratives and narratives are becoming increasingly quantitative."

PATHS FORWARD

The data we gathered indicate a strong demand for computing knowledge from students across a wide variety of disciplines. Looking forward, a central objective of the CS community, we believe, should be to better facilitate this learning in a way that supports all disciplines' computing interests, while encouraging their distinctive academic identities.

We foresee advantages for institutions requiring an introductory CS1 course across all — or a broad set of — majors. Similar to a writing or public speaking requirement, having a central starting point from which other disciplines can build is crucial to mitigating the "Tower of Babel" problem, in which superficial differences in communication prevent collaboration.

That said, a variety of CS1 flavors can open paths to making computing part of the identity of non-CS fields. Our first decade of experimentation with this approach demonstrates definitively that both CS and partnering disciplines benefit. In addition, while it is reasonable to maintain non-computing paths through many disciplinary specialties, we sense an emerging, bottom-up tension among students who want to leverage computing's mindset and skillsets without adopting a CS identity. This is where bridges from CS1 and injections into other courses offer particular promise: if all students have acquired a basic background in computing, it makes the process of using computing elsewhere more collaborative and natural: that "basic background" is the heart of the call to "CS for All."

As CSers, we relish our role as stewards of many facets of computation. Yet "CS for All" means much less ownership and much more fellowship with respect to computing. The call is to the varied and growing opportunities for collaboration and the multitude of widely-branching curricular paths, to which all disciplines will contribute. We look forward to the journey. It's a great era for computing!

ACKNOWLEDGMENTS

We acknowledge and thank the NSF for making this work possible, specifically through the support of projects #1612451 and #1659805. We also acknowledge the support of the Harvey Mudd College CS summer research program.

REFERENCES

[1] M. Zug, H. Hoffman, F. Kobayashi, M. President, and Z. Dodds. Supplemental Materials, CS35 https://www.cs.hmc.edu/~dodds/cs35/ (2018).

[2] J. Barnes and P. Hut. 1986. A hierarchical O(N log N) force-calculation algorithm. Nature 324 (Dec. 1986), 446–449.

[3] Valerie Barr. 2016. Disciplinary Thinking, Computational Doing: Promoting Interdisciplinary Computing While Transforming Computer Science Enrollments. ACM Inroads 7, 2 (May 2016), 48–57.

[4] Heather Bort, Mimi Czarnik, and Dennis Brylow. 2015. Introducing Computing Concepts to Non-Majors: A Case Study in Gothic Novels. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 132–137.

[5] Forrest J. Bowlick, Daniel W. Goldberg, and Sarah Witham Bednarz. 2017. Computer Science and Programming Courses in Geography Departments in the United States. The Professional Geographer 69, 1 (2017), 138–150.

[6] George Johnson. 2001. The World: In Silica Fertilization; All Science Is Computer Science. New York Times (Mar 2001).

[7] Darakhshan J. Mir, Sumita Mishra, Paul Ruvolo, Lori Pollock, and Sam Engen. 2017. How Do Faculty Partner While Teaching Interdisciplinary CS+X Courses: Models and Experiences. J. Comput. Sci. Coll. 32, 6 (June 2017), 24–33.

[8] Randall Monroe. 2017. Here to Help. (2017). https://xkcd.com/1831/

[9] John Peterson and Greg Haynes. 2017. Integrating Computer Science into Music Education. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 459–464.

[10] Steven A. Quigley. Academic identity: A modern perspective. Educate 11(1) (2011), 20–30.

[11] Grant Sanderson. 2017. manim. https://github.com/3b1b/manim.git. (2017).

[12] Nick Senske. 2017. Evaluation and Impact of a Required Computational Thinking course for Architecture Students. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 525–530.