CS + X Meets CS 1: Strongly Themed Intro Courses

Robert H. Sloan (Moderator)

University of Illinois at Chicago
Chicago, Illinois
sloan@uic.edu

Valerie Barr

Mount Holyoke College
South Hadley, Massachusetts
vbarr@mtholyoke.edu

Heather Bort

Marquette University
Milwaukee, Wisconsin
heather.bort@marquette.edu

Mark Guzdial

University of Michigan
Ann Arbor, Michigan
mjguz@umich.edu

Ran Libeskind-Hadas

Harvey Mudd College
Claremont, California
hadas@cs.hmc.edu

Richard Warner

Chicago, Illinois
rwarner@kentlaw.iit.edu

ABSTRACT

Typical CS 1 classes are *about* many things. The problems and examples are drawn from a variety of domains, with a goal of teaching a computational problem-solving approach and specific language constructs. Many CS 1 courses begin with writing programs that perform simple calculations, transition to more substantial simulations or games in the middle, and perhaps one or two SIGCSE Nifty Assignments at the end. In this panel, we explore the ‘strongly-themed’ approach in which many of the programs and examples used in the course come from a single domain which provides a theme for the course.

KEYWORDS

CS 1, CS + X

ACM Reference Format:

Robert H. Sloan (Moderator), Valerie Barr, Heather Bort, Mark Guzdial, Ran Libeskind-Hadas, and Richard Warner. 2020. CS + X Meets CS 1: Strongly Themed Intro Courses. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*, March 11–14, 2020, Portland, OR, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3328778.3366975>

1 SUMMARY

The SIGCSE community has created a number of strongly themed CS 1 courses, including Guzdial’s Media Computation course at Georgia Tech [5, 6], Harvey Mudd College’s “CS 5 Green” biology-themed introductory course [4] course, the multiple contextualized CS1s at Union College [1], and UIC’s law-themed course [8]. Most introductory computer science courses are taught with multiple contexts. The context or narrative around the programming examples or assignments are about mathematics (e.g., computing factorial), general science (e.g., converting between Fahrenheit and Celsius), or business applications (e.g., implementing bank accounts). A multiple-context CS 1 relies on students imagining how the concepts used in the homework might apply in future computational tasks of interest. In contrast, a *strongly-themed* or *contextualized*

course focuses on one or a small number of themes that connect all examples and homework assignments [6].

Today, strongly-themed introductory computing courses are becoming more common. A thread running through all of our presentations is that *a strong theme or context helps introductory CS students* in several different ways. For example:

- A context gives students a *reason* to do computing. A theme provides *concreteness*, clearly showing the value and meaning of solving problems of interest.
- For students taking CS 1 because of requirements (of a degree program or even parental pressure), a strong theme supports *engagement* and may even help convince them to continue with more CS.
- In spite of considerable efforts and some progress, computer science still struggles to draw women and students from traditionally underrepresented groups. Teaching CS 1 with contexts can *help promote diversity*. For example, biology and the humanities draw more women than men at the college level. CS 1 courses that leverage those interests have the potential to capture the imagination of students who might otherwise be disinclined to studying computing.
- CS enrollments continue to be extremely high, presenting both challenges and opportunities to most institutions. One opportunity is to use the concomitant increase in instructors to *teach a variety of themed CS 1 courses*, enhancing student engagement and resulting in better learning outcomes. Today, even many very small institutions will offer more than one CS 1 section per academic year, and can thus offer both strongly contextualized CS 1 *and* multiple-contexts CS 1 if desired.

The last point bears repeating: Today, even many small schools will offer multiple CS 1 sections per year, and thus can offer both a strongly contextualized *and* a multiple-contexts CS 1 if desired.

2 PANEL STRUCTURE

Each panelist will briefly give an overview of a strongly-themed CS 1 course at their institution, discussing teaching experiences, advantages, and drawbacks with respect to educational success, student interest and retention, and diversity.

3 VALERIE BARR

Valerie led the move to contextualized introductory computer science at Union College. The Union CS department launched six

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGCSE '20, March 11–14, 2020, Portland, OR, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6793-6/20/03.
<https://doi.org/10.1145/3328778.3366975>

different theme-based courses, each of which covered a common set of concepts and skills but appealed to a slightly different sector of the student population. Over a 10 year period this curriculum led to a significant increase in the number of students and a significant shift in the demographics of who took the introductory course.

Today this approach is being adopted at Mount Holyoke College. The themes introduced are slightly different than those at Union, reflecting different faculty interests and institutional priorities, but the demographic differences are equally striking. While Mount Holyoke will be in only its fourth semester of this new curriculum as of the time of SIGCSE, it will be possible to make preliminary comments on the appeal of this new curriculum to students who are unlikely to have otherwise taken an introductory computer science course.

4 HEATHER BORT

Heather has worked to change the way in which quantitative literacy is incorporated into the curriculum in several courses for English majors at Alverno College by introducing computer science concepts as the method for quantitative inquiry within those courses. Computing, as a tool for analysis and construction, was introduced to students that would not otherwise have been exposed.

The students were taking these courses as part of their normal path to a B.A. in English and computing was a tool used to provide additional value to their work within that context. We were challenged with the requirement of not losing any of the original content for the class and embedding CS content that was consistently built upon over the semester. While we are not able to cover all of the material that a student would learn in a CS 1 course over a single semester, that content is spread over a number of courses that English major students would take in a series.

5 MARK GUZDIAL

Mark developed Media Computation at Georgia Tech in 2002 because of a problem. Students in liberal arts, management, and architecture were failing or withdrawing their required computer science course at a rate of around 50 percent per semester. Media Computation was developed to improve success rates. Media Computation students write functions that loop through all the pixels of a picture to implement Photoshop-like filters and all the samples of the sound to create sound effects. The result was a decrease in the failing or withdrawing rate to about 15 percent [7]. A series of studies over ten years suggests that Media Computation was successful at improving course success rates because the consistent theme of digital media helps students understand the *relevance* of the computing to their lives.

6 RAN LIBESKIND-HADAS

Ran Libeskind-Hadas co-developed and regularly co-teaches “CS 5 Green” at Harvey Mudd College. This is a version of Harvey Mudd’s introductory CS course in which the topics and problems are motivated by applications in biology. Some of the important features of this course are: (1) The course covers all of the CS material of the college’s “standard” introductory course, (2) students write their own code, from scratch, that allows them to analyze data and make biological discoveries from those data, and (3) the biology content

is self-contained. The course has been taught at Harvey Mudd since 2009 and has been adopted and adapted at several other colleges and universities, several with student bodies that are diverse in their backgrounds and interests. Interestingly, a larger fraction of graduates of this course choose to major in CS than their peers in the “standard” introductory course. Moreover, a very significant fraction of the college’s Biology as well as Mathematical and Computational Biology majors are graduates of this course.

7 ROBERT H. (BOB) SLOAN

Bob has, with colleagues, developed and introduced both a CS + Law intro (see next section), and a version of the Harvey Mudd CS + Bio intro that is more suitable for students at UIC, a typical selective large public institution, as opposed to the students’ of one of the most selective small schools in the country. Initial results suggest strong student engagement and success, but (so far), not dramatically different demographics [2, 8].

8 RICHARD WARNER (NON-COMPUTER SCIENTIST TEAM TEACHER)

Bob and Richard have co-taught a CS 1 class for three years. Having a law professor in the classroom fulfills an ambition of almost every CS department: to have students engage with relevant policy issues. Just as “[c]omputation is a widely accepted lens for looking at the world” [3], law is a lens for looking at policy issues in computer security and privacy. Integrating the perspectives brings policies issues alive in a way that can be difficult to achieve otherwise and also deepens students’ interest in the coding projects themselves.

ACKNOWLEDGMENTS

This panel is based on work supported by the National Science Foundation under Grants No. DUE-1612113 (Sloan), DUE-1612455 (Sloan and Warner), IIS-0722203 (Barr), and 0618674 & 0634629 (Guzdial).

REFERENCES

- [1] Valerie Barr. 2016. Disciplinary Thinking, Computational Doing: Promoting Interdisciplinary Computing While Transforming Computer Science Enrollments. *ACM Inroads* 7, 2 (May 2016), 48–57. <https://doi.org/10.1145/2891414>
- [2] Tanya Berger-Wolf, Boris Igic, Cynthia Taylor, Robert Sloan, and Rachel Poretsky. 2018. A Biology-themed Introductory CS Course at a Large, Diverse Public University. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM Press, 233–238. <https://doi.org/10.1145/3159450.3159538>
- [3] Peter J. Denning. 2009. The Profession of IT: Beyond Computational Thinking. *Commun. ACM* 52, 6 (2009), 28–30. <http://dl.acm.org/citation.cfm?id=1516054>
- [4] Zachary Dodds, Ran Libeskind-Hadas, and Eliot Bush. 2010. When CS 1 is Biology 1: crossdisciplinary collaboration as CS context. In *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '10)*. 219–223. <http://doi.acm.org.proxy.cc.uic.edu/10.1145/1822090.1822152>
- [5] Mark Guzdial. 2003. A Media Computation Course for Non-Majors. In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '03)*. 104–108. <https://doi.org/10.1145/961290.961542>
- [6] Mark Guzdial. 2010. Does Contextualized Computing Education Help? *ACM Inroads* 1, 4 (Dec. 2010), 4–6. <https://doi.org/10.1145/1869746.1869747>
- [7] Mark Guzdial. 2013. Exploring Hypotheses About Media Computation. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research (ICER '13)*. ACM, New York, NY, USA, 19–26. <https://doi.org/10.1145/2493394.2493397>
- [8] Robert H. Sloan, Cynthia Taylor, and Richard Warner. 2017. Initial Experiences with a CS + Law Introduction to Computer Science (CS 1). In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM Press, 40–45. <https://doi.org/10.1145/3059009.3059029>