A New Model for Weaving Responsible Computing Into Courses Across the CS Curriculum

Lena Cohen
lena_cohen1@brown.edu
Brown University
Providence, Rhode Island, USA

Heila Precel
heila.precel@gmail.com
Brown University
Providence, Rhode Island, USA

Harold Triedman
harold_triedman@brown.edu
Brown University
Providence, Rhode Island, USA

Kathi Fisler
kfisler@cs.brown.edu
Brown University
Providence, Rhode Island, USA

ABSTRACT

CS departments in the USA have used various models for teaching about ethics, including standalone ethics courses and expert-designed assignments. At Brown University, we are trying a different approach: a group of undergraduate teaching assistants dedicated to socially-responsible practices in computing work with faculty to integrate content into multiple assignments both across a course and across the curriculum. This "responsible computing" initiative has resulted in a variety of assignments added to 13 different courses in the past year. This paper describes the program's design, sample assignments, results of an internal evaluation, and refinements we are making to the model. We contrast our model to others from the literature in hopes of expanding the collection of curricular ideas for designing education in responsible computing.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**; • **Human-centered computing**;

KEYWORDS

CS Ethics; Responsible Computing; University CS Curricula; Teaching Assistants

ACM Reference Format:

Lena Cohen, Heila Precel, Harold Triedman, and Kathi Fisler. 2021. A New Model for Weaving Responsible Computing Into Courses Across the CS Curriculum. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432456

1 INTRODUCTION

Computer science educators have called for including ethics in the curriculum since the first SIGCSE conferences in the early 1970s [3]. Unfortunately, these efforts have not penetrated the broader sphere of public discourse, leading to criticism [10], public backlash [16], and general distrust of the technology industry [17]. As departments reckon with these outcomes, more are looking for models that will be appropriate and effective within their contexts.

This paper describes the model that we are developing in the CS department at Brown University, a highly-selective private university in the northeastern USA. Our model differs from other publicized models, including standalone ethics courses and Harvard's multi-course Embedded EthiCS project [5]. We designed our initial model in Spring 2019, used it throughout the 2019-2020 academic year. In spring 2020, we conducted an internal evaluation based on interviews and focus groups with students, teaching assistants, and faculty. We've made several refinements for the current (2020-21) academic year based on our findings.

Two features of our model distinguish it from others in the literature. First, our focus—which we term *Socially-Responsible Computing*—is broader than ethics: we include a combination of understanding power and technology, making ethical design decisions, building accessible systems, and testing systems for (un)desirable impacts on various stakeholders. We believe the broader framing and terminology matter in practice. "Ethics" is a topic that students and faculty associate with the Philosophy department: CS faculty feel unqualified to teach it and some students see it as out of the scope of computing. At least in the USA, the field can suggest privilege, seeming detached from the communities most harmed by irresponsible computing.[1] We strive to widen the term and the umbrella to make Socially-Responsible Computing an essential topic from the outset in the CS curriculum. As a second novel feature, our model makes extensive use of *undergraduate teaching assistants* for the design and deployment of materials. We find many undergraduates are passionate about addressing the current social impacts and failures of technology. They are a rich resource for identifying possible materials and ideas for communicating them to fellow students. Given support from content and pedagogy experts, students can add significant value while expanding their own learning.

Section 2 describes historical and recent related efforts. Section 3 describes our model in detail. We present sample assignments developed under our model in section 3.1. Section 4 presents the internal evaluation that we conducted in the second semester of the effort,

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432456

[1] A researcher in justice and online behavior called out Harvard's program on this on Twitter, saying "there's little Black or disability scholarship (but there's Mills, Kant, and Rawls!). Either ethics needs to be broader or its only one part of this" [14].

with revisions to the model in section 4.2. Section 5 discusses key observations that might be of use to other CS departments.

2 RELATED WORK

The history of CS ethics education highlights several general models. Movement toward integrating ethics into CS curricula began in earnest in 1978. That year, the ACM's new curriculum proposed an elective course entitled "Societal Impacts of Computing" [9] and Maner wrote a "starter kit" for teaching computer ethics [1]. By the mid-1980s, Pecorino [11] proposed a standalone course in computer ethics. Several years later, Miller [9] sought to show that it was possible to integrate ethics into any CS class by writing numerous examples of ethical dilemmas for various classes.

While inclusion of some ethics content is common across CS departments, the nature of that coverage varies. In 2006, Quinn [12] surveyed the course offerings of nearly a quarter of accredited CS departments in the USA and found that more than half offered a "social and ethical implications of computing" course taught by a computer science faculty member. An additional fifteen percent of departments met accreditation requirements by having students take classes in another department, usually philosophy. In 2011, Goldweber et al. [4] similarly found that "only a few schools (5%) do not cover social, ethical, and/or professional issues [in their computing curriculum]." However, the depth of engagement with computing ethics and its pedagogical context are highly inconsistent. In 2020, Fiesler et al. [3] reported heterogeneity across a survey of hundreds of tech ethics curricula, encountering consistency only in the desire to "teach students to recognize ethical issues in the world, critically evaluate and assess these issues and technologies, and make well-reasoned arguments based on these critiques."

Three efforts are particularly relevant to our proposed model.

ImpactCS. ImpactCS [6–8] began in the 1990s. Martin et al. established learning objectives and "knowledge units" as a framework for teaching and learning computing ethics. They also proposed a "three-dimensional knowledge space" with which to analyze subdisciplines of CS from various ethical perspectives. Ultimately, their goal was to underscore that "every ethical concern is located at a particular level of social analysis. Only an analysis that takes account of at least three dimensions–the technical, the social, and the ethical–can adequately represent the issues as they concern computer science in practice." The project's recommendations draw on theories of applied ethics, learning, and moral education.

ImpactCS does not prescribe a rigidly-defined, separate structure for ethics education. Rather, the project advocates for progressive integration of social and ethical content across the curriculum, maximizing coverage of ideas while minimizing overlap. Our program shares these goals. The model we present in this paper goes a step farther, exploring specific implementation decisions for a cross-cutting educational program in responsible computing.

Embedded EthiCS: Integrating EthiCS across CS Education. The Embedded EthiCS program at Harvard [5] incorporates and distributes ethics education across many courses in the CS department. Philosophy graduate students and postdocs collaborate with CS faculty to create ethics modules for courses in the core curriculum. Its design reflects a belief that "[u]nderstanding, evaluating, and successfully defending arguments about what should be done falls within the purview of the normative disciplines, most notably ethics, a subfield of philosophy." Each Embedded EthiCS module connects an ethical issue to relevant philosophical principles and theories; the module spans 1-2 class meetings and a corresponding assignment. At time of this writing, the Embedded EthiCS website listed activities connected to 17 courses, including data science, architecture, networks, programming languages, and robotics. Our effort focuses more broadly on responsible computing as a design, implementation, and validation practice, rather than on theoretical ethical underpinnings. The Embedded EthiCS modules present isolated ethical content within courses, while ours weave content deeply into technical course assignments. A single course in our model may touch on multiple responsible-computing topics, often engaging multiple perspectives and development phases. Our program also relies on undergraduate teaching assistants, rather than experts (e.g., postdocs) from disciplines like philosophy. Section 3 discusses the implications of this staffing decision.

Integrating Ethics Into Multiple Courses. Two teams with overlapping authors describe efforts to integrate ethics education continuously across CS courses in HCI [15] and Machine Learning (ML) [13]. The HCI course stresses ethical thinking; it incorporates ethics through guest lectures, in-class activities, and assignments consistently throughout the course, with each encounter building on those that came before. The authors emphasize the need for ethics education to include practical training on how to weigh the consequences of decisions. The course grounds this training in the legal, psychological, and historical dimensions of technology. The course includes readings, discussions, and answering written questions. In the ML course, the instructors created a framework to help students identify ethical issues within an ML project, focusing specifically on issues that are "actionable by members of an ML project team, and not those that are societal in nature." They use this framework to create ethical reflection questions around concepts that are already common in ML classes (such as logistic regression, random forest classifiers, and convolutional neural networks). The authors emphasize the importance of practicing the mechanics of ethical thinking even when working with common datasets, which motivates the need for the explicit framework. Our project works with a larger range of courses, which in turn raises a wider range of issues (such as the design, implementation, or analysis of social concerns surrounding computing systems).

3 THE BROWN UNIVERSITY MODEL

The Brown CS initiative on socially-responsible computing began in 2018. The coursework component, which is the focus of this paper, began in early 2019. It aims to include relevant, societal-focused, and practice-oriented content across the (undergraduate and early graduate) curriculum. The program's mission statement lists three goals for our students:

(1) recognize that technology is not neutral, nor does it exist in a vacuum: it contains built-in biases that reflect the preferences, norms, and worldview of its creators
(2) build with everyone in mind
(3) understand and fulfill the responsibility to advocate against unethical product or research decisions

The program does not prescribe how individual courses should work towards these objectives. Each course's staff decides on its content and mechanisms (e.g., presented in lecture, written questions on assignments, separate readings, labs, part of course projects, etc). Each course is expected to touch on responsible computing multiple times during the semester, at least once as part of a graded assignment, with coverage starting earlier in the semester (rather than as an add-on at the end). Our near-term intent is that all majors will take 4-8 courses that are enhanced with such materials.

To assist with identifying and integrating content into courses, we decided to hire undergraduate teaching assistants (TAs) to help with identifying and integrating content into courses and assignments. Relying on undergraduate TAs has long been part of the culture of our department. In fact, our courses run almost entirely on undergraduate TAs (with graduate TAs working mainly on specialized upper-level courses for which undergraduates are not available). A majority of our undergraduates serve as TAs at least once during their time at Brown; many take on the role multiple times. Each course has 1-4 "head" TAs who manage the rest of the TAs while working more closely with the professor. Head TAs frequently propose assignment designs and initial grading rubrics, so this practice was well within departmental norms.

From the outset, we knew we would need colleagues with expertise in ethics and social sciences to help us identify and frame materials. Despite trying, we were initially unable to find someone on campus with both suitable expertise and availability. We therefore drew on the somewhat scattered knowledge of faculty and students along with the resources of the internet for the first year; a content expert joined the team in Year 2 (see section 4.2).

Logistics: We created a new class of "socially responsible computing TAs" (henceforth STAs). Faculty may request to have two STAs assigned to their course. These are paid positions, typically 6-8 hours of work per week for the semester. The nature of an STA's work varies from course to course: some professors already have ideas for content or assignments that they ask the STAs to implement (drafting handouts and webpages, and sometimes grading rubrics). Others rely on STAs to generate ideas. In most courses, STAs also help with the grading of responsible-computing assignments (undergraduates are allowed to grade assignments at Brown).

The department chair oversees the program. Two head STAs (paid undergraduates) coordinate hiring for and manage the direction of the STA program. The program looks for applicants with previous academic exposure to relevant topics, assignment-development experience (e.g., prior regular TA experience), or demonstrated interest in the social aspects of computing. So far, all STAs have been undergraduate CS majors, but many STAs are double majoring in a non-STEM subject (such as History, Science and Society, American Studies, Comparative Literature, or Modern Culture and Media). The STAs meet as a group throughout the semester to discuss ideas and share notes about course assignments.

3.1 Courses and Representative Assignments

Four courses participated in the program in Fall 2019 and ten courses participated in Spring 2020 (one participated both times). These included multiple introductory-level courses, software engineering, security, machine learning, deep learning, data science, user-interface design, and web application design. In addition, the programming languages professor added content without working with an STA. A separate upper-level (theory) course on social implications of cryptography was also introduced outside of the STA program. All in all, the participating courses occur across the four years of the undergraduate curriculum (first year graduate courses are the same as the upper-level undergraduate courses). Over a quarter of the department's faculty have been involved thus far.

Table 1 summarizes how a selection of these courses incorporated responsible computing content.[2] One column describes the topics that were included, while another describes the format in which the content was integrated. The table illustrates the systems focus of many of these projects, with some focused on design, some implementation, and some methods for testing or validating systems for adverse impacts on different stakeholders.

4 EVALUATION AFTER YEAR 1

In spring 2020 (the second semester of the program), we conducted an evaluation to check whether the program was operating effectively from the perspective of each of STAs, students, and faculty. We wondered whether participants agreed on the program goals, how goals were (not) being met, and what revisions to make going forward. The evaluation was conducted as part of an undergraduate research project. The student researchers (the first three authors, all of whom were also STAs) worked closely with a faculty advisor (the fourth author, who has expertise in computing education and qualitative methods) to design the study questions. The methods involved focus groups with each of (non-STA) students and other STAs, as well as interviews and surveys with CS faculty both within and outside the program. A total of 12 students participated in one of four student focus groups, while 11 STAs participated in the three STA focus groups (representing seven courses, though none at the introductory level). Five faculty who had worked with STAs gave interviews, and another seven responded to the faculty survey. All focus groups were conducted through videoconferencing in April 2020 (after campus closed for COVID).

In terms of the questions:

- Student focus groups discussed definitions of socially responsible computing, what topics are important for students to see, and what activities students were seeing in courses they were taking and which seemed effective or ineffective. We also asked them to talk about comments they were hearing from friends about the content being added to courses.
- The STA focus groups started with similar questions to the student groups, then turned to the content-development process. STAs were asked how they chose topics and resources, who their target audiences were, and the ways in which they worked with the course professor in this work.
- Faculty were asked about the responsible-computing topics that were relevant to their course topics, what constraints they faced in integrating the new content, what goals they had for students who engaged with the material, and how they thought about assessing the new materials.

We acknowledge that our use of voluntary focus groups naturally biased our student population: most student participants were

[2] Details and more examples are online at https://ethics.cs.brown.edu

Table 1: Samples of courses and assignments.

Course Title	Responsible Computing Topics	Content Format
Intro Computing (CS1)	Raised different ways to match advertisements to users, limits of making decisions around available datasets, and potential inferences from metadata.	Short lecture discussions. A progression of methods for matching ads to users across multiple programming assignments. Short readings and written reflections associated with many homeworks. One course project augmented with a reading and reflection connected to the project theme.
Data Structures (CS2)	Introduced the concept of a socio-technical system and framed technical problems within real world social contexts. Highlighted ethical and social issues surrounding algorithms and data structures. Topics included ML bias, photo manipulation, and the spread of disinformation through search results.	Updated all major projects to include a social context and written ethics questions. Created a new dataset and classification task for the decision tree project to focus on algorithmic hiring bias. Added questions to two homeworks.
User Interfaces	Taught students about implicit value judgements that come with design choices. Practiced considering ethics during the design process. Topics included Microsoft Guidelines for Human-AI interaction, unethical metrics for A/B testing, dark patterns, accessibility, and bias in sampling.	Embedded questions in each assignment. Incorporated content in lecture and led in-class activities. Created an assignment where students choose ethical values for assessing interfaces and create one model that aligns with these values and one that operates against them.
Deep Learning	Explored the ethical implications of deep learning technologies that map directly to course concepts as well as broader systems-level concepts. Topics included interpretability, fairness, energy consumption, privacy, and deepfakes.	Incorporated ethics content in lecture. Incorporated written ethics questions in every assignment. Added lab about reducing gender bias in natural-language models.
Security	Considered the implications of systems security from the perspective of users, software developers, and policymakers. Assignments explored issues with digital lockpicking, responsible vulnerability disclosure, internet shutdowns, TOR and the Dark Web, internet censorship, and cryptocurrency.	Incorporated responsible-computing content in lecture. Developed a corresponding component for every homework, which included reading 1-5 articles and answering 3-5 questions.

interested in the program and in seeing it succeed (even though none were themselves STAs). For this round of formative evaluation, this limitation is appropriate as we wanted to identify how things were working at the overall structural level.

We did not look at data from end-of-semester course evaluations for information about the responsible-computing assignments. Our focus was on the overall program across the courses, not the details within individual courses. In addition, spring course operations had been sufficiently disrupted by COVID that we wouldn't have been sure how to interpret certain responses.

4.1 Analysis and Results

We transcribed the interviews and used open-coding to analyze the results. While many themes aligned to our questions, several interesting observations arose that we had not asked about. The following discussion groups observations by theme.

4.1.1 Perspectives on responsible computing as a discipline. Participants of all three types (faculty, students, and STAs) pointed out that technological artifacts embody an ethics and can serve as a source of injustice (with special emphasis on technology abuse for surveillance and control) and that there should be equitable access to technological artifacts across social groups. Participants highlighted that responsible computer scientists must think through unintentional ramifications of all technical projects in which they participate. Many students and STAs (but only one professor) specifically framed responsible computing as professional decency within the tech industry.

4.1.2 Metrics for program success. Students, STAs, and faculty all felt that a successful program would lead to students effectively actualizing their (own) values in future professional decision making. In the words of a student: "I think the real measure of success isn't just to ask people to reflect on having learned something. I think the real measure would actually be asking people to identify times where something they learned in CS ethics affected their choices or their behaviors." An STA remarked that students graduating from Brown CS should "identify the ways in which technology holds and directs power," take ownership for code they produce, and "ask the questions that people aren't asking." Some student participants also wanted their peers to see that "the ethics stuff is cool." Faculty wanted students to understand the social context of their work and act in accordance with ethical and responsible principles.

4.1.3 Evaluation of assignment efficacy and student learning. Students' favorite responsible-computing assignments either started "cool conversations" among their friends or challenged them intellectually. Students wanted future assignments to "[make responsible-computing content] more spontaneous and more of a treat". Several suggested keeping the new content out of technical assignments to allow more depth in the treatment of responsible-computing content and to prevent classmates from complaining about "tacked on" and "forced" questions. Outside of the study, however, some faculty reported hearing the opposite request from students who believed integration was essential for making the content relevant.

STAs perceived the integrated technical assignments as both more effective and more popular. STAs were consistently surprised by the level of student engagement with written questions, while also acknowledging that students could get away with not taking the content too seriously. They also felt that written assignments were important for "big picture" social and structural questions, but acknowledged that these were hard topics to cover adequately within the confines of a single course. One faculty member raised concerns that written assignments fell short in avoiding the back-and-forth discussions that some saw as necessary for processing such content. These comments suggest that a standalone course would be a useful complement to the integrated content. Section 5 returns to this point.

One student expressed concern that more direct instruction was needed in some assignments, saying "I don't know enough about ethics to criticize a code of ethics. I'm not good at thinking of criticisms against stuff ... I almost wish there was more of an instruction piece, I guess, as opposed to a philosophical piece."

Many STAs and faculty, not trained in computing education, viewed engagement as a proxy for assignment efficacy when interpreting student feedback within their courses. Some professors struggled to figure out how to grade questions on topics that were inherently holistic, given the contrast to the more familiar forms of technical assignments. We return to this question in section 5.

4.1.4 Perceived legitimacy of responsible-computing content. In the pilot year, many courses left their lecture content intact and put the responsible-computing content into labs, assignments, and projects, which were less time-constrained. For some students, this lessened the perceived importance of the content. "Students look to professors as role models," said one student, "so if it's taught to [me] as [an important concept in lecture], it feels like my professor, who I respect a lot, wanted me to learn this." Another observed, "I find that when we're behind the scenes, trying to do an ethics problem, it's all jokes. But when we're in class and an ethics problem comes up, everyone starts looking and paying more attention and definitely sitting up." Students also raised concerns that tacking responsible-computing content onto the end of an assignment led students to rush through it.

4.1.5 Perceptions of content as political. Both STAs and students worried that other students might perceive the new content as politically charged and thereby dismiss it. To counter this, STAs reported avoiding terminology that, while appropriate, might raise negative responses for some students. One student said, "[I worry that some people would perceive required ethics as] the university trying to get us to all think the same way, just like all the liberals here think. And so I worry about pushing away people who probably need to think about ethics more but who are just sort of resistant to it because of certain associations." Interestingly, although focus-group participants pointed to friends who thought that ethics content was unnecessary, none could cite examples of students who felt that the content was too political. While not all social risks of systems have political overtones (e.g., thinking about traffic routing around playgrounds), finding such examples takes effort, especially when STAs might be pre-disposed to thinking about headline issues instead.

Some STAs also mentioned the need to monitor content ideas to consider potentially sensitive issues for different student demographics or cultures. One explicitly mentioned deciding to steer clear of a discussion about privacy and surveillance in China, out of respect for the large population of Chinese students taking the course they had worked on.

4.1.6 Expertise and content development. While many faculty in the CS department are eager to work more responsible-computing content into our curriculum, few have expertise in this area. Many professors who requested STAs for their courses did so specifically because they felt unqualified to implement such content themselves. Some had learning goals and thematic ideas that they wanted to communicate to students, while others had specific project ideas that they wanted to actualize. However, most expected that STAs—who would bring in their own personal interests in these issues—would lead on content design, implementation, and evaluation. As one professor said, "The fact [is] that [ethics] is an age-old subject about which most CS profs have only very general knowledge." STAs, on their end, lacked the broader expertise needed to make up for gaps in faculty knowledge: they too wanted more formal training in social sciences and technology. This shared lack of expertise stood out particularly in courses that strove to add a responsible-computing component to nearly every assignment. In addition, even when STAs had strong content ideas, they reported that their lack of training in pedagogy made them hard to implement. This is hardly surprising, yet it highlights a challenge when neither the faculty nor the TAs feel completely in command of the topic at hand, despite shared enthusiasm to cover the material.

4.2 Modifications for Year 2

Year 2 of the program began in Fall 2020 with several changes. An advanced PhD student from the Modern Culture and Media program joined the team as a content consultant. He helps STAs and faculty identify materials and key themes to raise, while also training STAs and interested faculty in developing reading and reflection assignments; he is paid for four hours per week throughout the academic year. Co-author Fisler, a CS faculty member with expertise in computing education, joined the team to guide the STAs on cross-course coordination and general pedagogic planning; she is spending roughly 8 hours per semester as part of her departmental service load. A PhD-level staff member from the university's Center for Teaching and Learning is training STAs on articulating pedagogic objectives and designing materials towards them. STAs are also working with the department's diversity committee and student diversity consultants (a couple of hours per semester) to identify content ideas and sanity check assignments for unintended adverse consequences. While the STAs retain significant agency in the direction of the program, these new partnerships should strengthen and deepen what we achieve.

We are also doing some cross-course analysis of assignment content and themes, working with our PhD-student expert consultant to frame both technical and social learning objectives across the 4 years of the curriculum. Our current technical objectives are *recognize and understand the social context of a problem*, *design for* *everyone, not just typical users*, and *test algorithms or systems for bias or potential harm*. The social-impact objectives (which are still evolving) include *recognizing that both code and data can encode social assumptions*, *creating systems for accountability*, and *avoiding technological solutionism* (among others).

We have also realized that a promising feature of our model lies in the education that it affords the STAs themselves. They learn new content, learn about pedagogy, and get feedback from both faculty and students on the effectiveness of their ideas. We are beginning to gather data that will help us study the nature of the impact on the STAs over time. Both this and the cross-course assignment analysis will be conducted by undergraduate research assistants working with the program's faculty leadership.

5 TAKEAWAYS FOR OTHER DEPARTMENTS

Our evaluation and ongoing discussions among STAs and faculty suggest several broader lessons that extend beyond our department.

Lesson 1. *A standalone course on the societal impacts of computing remains valuable, alongside extensive embedded content.*

Each of faculty, STAs, and students have raised concerns that some important topics don't fit well within individual courses. Fundamental theories from the social sciences and ethics are high on this list (Connolly's recent CACM article [2] argues for this in more detail). A standalone course would also provide more opportunities for both deeper discussion and projects that integrate multiple socially-responsible concerns: in an embedded setting, it is harder to make space for projects that include both universal design and fairness testing, for example, since the projects also have to exercise other core content. The embedded format remains valuable, however, for exposing all students to some of this content in context, as a way of modeling professional practices that include responsible decision-making. Long term, we expect to use both formats.

Lesson 2. *Expertise is needed in pedagogy and instructional design, as much as around ethical and social content.*

While we had anticipated needing expertise in ethics and social science, we underestimated the need for expertise in pedagogy and instructional design. CS faculty generally lacked knowledge on how to frame questions to encourage meaningful engagement with readings, as well how to create grading rubrics for such questions. Holding engaging discussions in courses with over 200 students was another challenge. Fortunately, pedagogy expertise is more readily available than technology and social science expertise on many campuses, in the form of Teaching Centers or local faculty experts on education and pedagogy. Even as more assignment ideas emerge online regarding social sciences and technology (as has been happening at least on social media), the needs for expert advice on pedagogy and assessment are likely to remain.

Lesson 3. *Undergraduate TAs bring some genuine strengths when weaving socially-responsible-computing content into courses.*

Undergraduates have an ability to connect with their peers in many ways: they know what issues students are thinking about, they hear more grapevine about kinds of assignments that are and aren't working, and they have deep knowledge of how individual courses and their assignments work from a student perspective. If the STAs reflect the diversity of the department, they can help identify ideas for locally-inclusive content. Finally, from a practical perspective, they exist in all departments that teach undergraduates (unlike postdocs or research scientists, for example), at the relatively modest costs of TA positions within the department.

We acknowledge that our department was primed for such an experiment to work given our long history of having undergraduate TAs help with course design. We view such work as part of the educational value to the TAs themselves, and this is content that many students (at least at our liberal-leaning university) want to learn. In using undergraduates rather than post-PhD experts, our content may take longer to reach maturity; the collaboration between faculty and students to get there, however, should build a stronger foundation in these issues across the faculty.

Lesson 4. *Having CS faculty present responsible-computing content in lectures helps students take the content seriously.*

Students in CS programs sometimes brush off non-technical material; anecdotally, some students have asked whether responsible computing is actually important since their internships don't ask them to practice it. Some focus group students cited the importance of hearing faculty raise these issues (assuming a setting in which students respect and trust their faculty). This message may also need to be communicated to faculty who are uncomfortable discussing such material due to feeling unqualified in these topics. This is another place where we expect to leverage our broader team: our new content expert and teaching consultant can help faculty plan and prepare discussions on responsible-computing topics.

Lesson 5. *Coordinating content discussions across courses is important to reduce repetition.*

We experienced two particular kinds of repetition of responsible-computing content across courses. Many introductory-level courses created assignments on algorithmic bias (alongside the machine learning courses), often using the same articles. In addition, multiple courses chose to have students read and reflect on the ACM Code of Ethics. In some cases, neither the faculty nor the STAs had strong ideas of what responsible-computing topics were relevant to their courses. Each hoped the other would generate ideas, and these repeated resources ended up as natural defaults. Our new team members, focused on coordination and content ideas, should address these issues. Our content expert is helping us assemble mini "catalogs" of materials that could be relevant to several standard computing topics: our goal is to provide these as resources for STAs as they select specific materials.

In the bigger picture, learning progressions that identify and stage dependencies across socially-responsible computing topics would also help. Computer science faculty already understand curricular progressions for technical content. Such a progression for the responsible-computing content would be a valuable addition to the literature. For the time being, our department is leveraging ideas about how we position programming content across courses to start such an effort (as briefly described in section 4.2). How to do this in ways that accommodate the range of theoretical and applied topics that make up modern computing (and data!) science curricula remains an open question for future work.

ACKNOWLEDGEMENTS

The Brown CS Department Chair, Ugur Centintemel, has provided vision and critical leadership of this program from its inception. Thanks to the STAs for their work and interest in this program, our focus group participants, and the many Brown CS faculty who have embraced this effort on a departmental scale.

REFERENCES

[1] Terrell Bynum. 2018. Computer and Information Ethics. In *The Stanford Encyclopedia of Philosophy* (summer 2018 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University, Stanford, CA.

[2] Randy Connolly. 2020. Why Computing Belongs within the Social Sciences. *Commun. ACM* 63, 8 (July 2020), 54–59. https://doi.org/10.1145/3383444

[3] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What Do We Teach When We Teach Tech Ethics?: A Syllabi Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 289–295. https://doi.org/10.1145/3328778.3366825

[4] Mikey Goldweber, Renzo Davoli, Joyce Currie Little, Charles Riedesel, Henry Walker, Gerry Cross, and Brian R. Von Konsky. 2011. Enhancing the Social Issues Components in Our Computing Curriculum: Computing for the Social Good. *ACM Inroads* 2, 1 (Feb. 2011), 64–82. https://doi.org/10.1145/1929887.1929907

[5] Barbara J. Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: integrating ethics across CS education. *Commun. ACM* 62, 8 (July 2019), 54–61. https://doi.org/10.1145/3330794

[6] C. Dianne Martin, Chuck Huff, Donald Gotterbarn, and Keith Miller. 1996. A framework for implementing and teaching the social and ethical impact of computing. *Education and Information Technologies* 1, 2 (June 1996). https://doi.org/10.1007/BF00168276

[7] C. Dianne Martin, Chuck Huff, Donald Gotterbarn, and Keith Miller. 1996. Implementing a tenth strand in the CS curriculum. *Commun. ACM* 39, 12 (Dec. 1996), 75–84. https://doi.org/10.1145/240483.240499

[8] Dianne Martin, Chuck Huff, Donald Gotterbarn, and Keith Miller. 1996. Curriculum guidelines for teaching the consequences of computing. In *Proceedings of the symposium on Computers and the quality of life - CQL '96*. ACM Press, Philadelpia, Pennsylvania, United States, 73–85. https://doi.org/10.1145/238339.238376

[9] Keith Miller. 1988. Integrating Computer Ethics into the Computer Science Curriculum. *Computer Science Education* 1, 1 (Jan. 1988), 37–52. https://doi.org/10.1080/0899340880010104

[10] Cathy O'Neil. 2017. Opinion | The Ivory Tower Can't Keep Ignoring Tech. *The New York Times* (Nov. 2017).

[11] Philip A. Pecorino and Walter Maner. 1985. A Proposal for a Course on Computer Ethics. *Metaphilosophy* 16, 4 (Oct. 1985), 327–337. https://doi.org/10.1111/j.1467-9973.1985.tb00179.x

[12] Michael J. Quinn. 2006. On teaching computer ethics within a computer science department. *Science and Engineering Ethics* 12, 2 (June 2006), 335–343. https://doi.org/10.1007/s11948-006-0032-9

[13] Jeffrey Saltz, Michael Skirpan, Casey Fiesler, Micha Gorelick, Tom Yeh, Robert Heckman, Neil Dewar, and Nathan Beard. 2019. Integrating Ethics within Machine Learning Courses. *ACM Transactions on Computing Education* 19, 4 (Nov. 2019), 1–26. https://doi.org/10.1145/3341164

[14] Sarita Schoenebeck. 2020. Twitter Post. https://twitter.com/syardi/status/1298939006550409218.

[15] Michael Skirpan, Nathan Beard, Srinjita Bhaduri, Casey Fiesler, and Tom Yeh. 2018. Ethics Education in Context: A Case Study of Novel Ethics Activities for the CS Classroom. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Baltimore Maryland USA, 940–945. https://doi.org/10.1145/3159450.3159573

[16] Kara Swisher. 2018. Opinion | Who Will Teach Silicon Valley to Be Ethical? *The New York Times* (Oct. 2018).

[17] Shoshana Zuboff. 2020. Opinion | You Are Now Remotely Controlled. *The New York Times* (Jan. 2020).