Reflecting on Practices to Integrate Socially Responsible Computing in Introductory Computer Science Courses

Kevin A. Wortman
kwortman@fullerton.edu
California State University, Fullerton
Fullerton, California, USA

Aakash Gautam
aakash@pitt.edu
University of Pittsburgh
Pittsburgh, Pennsylvania, USA

Sarah Hug
sarah.hug@colorado.edu
Colorado Evaluation & Research Consulting
Westminster, CO, USA

Paul Salvador Inventado
pinventado@fullerton.edu
California State University, Fullerton
Fullerton, California, USA

Ayaan M. Kazerouni
ayaank@calpoly.edu
California Polytechnic State University
San Luis Obispo, California, USA

Jane Lehr
jlehr@calpoly.edu
California Polytechnic State University
San Luis Obispo, California, USA

Kanika Sood
kasood@fullerton.edu
California State University, Fullerton
Fullerton, California, USA

Zoë Wood
zwood@calpoly.edu
California Polytechnic State University
San Luis Obispo, California, USA

Abstract

Socially Responsible Computing (SRC) education entails the infusion of Computer Science (CS) education with interwoven attention to ethical, social, and political issues to position students to reflect and take action individually and collaboratively to create a more just world. Our approach to SRC supports students to explore computing design/development in early CS courses with a communal goal orientation (in contrast to agentic/individualized), shown to improve achievement and retention for students with identities that are minoritized in CS. Grounded in our own experiences as co-developers and implementers of this pedagogical transformation and as co-facilitators of a Faculty Learning Community (FLC) across six minority-serving institutions in California, we share how we use an iterative design and implementation process modeled from social design experimentation as research and development method. Initial results are presented as a set of promising practices for incorporating SRC into introductory CS courses: 1) choose the domain mindfully; 2) design for synergy with technical material; 3) scaffold for inclusivity; 4) structure with a framework; 5) avoid othering SRC elements; and 6) reuse and build on existing resources. We share how these promising practices guide our efforts; how they can address challenges and concerns for new and continuing SRC implementers; and the ways in which we have and will continue to test and co-design this approach.

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02
https://doi.org/10.1145/3641554.3701962

CCS Concepts

• **Social and professional topics → Computing education**.

Keywords

CSEd, responsibility, social impact, ethics, computing education

ACM Reference Format:
Kevin A. Wortman, Aakash Gautam, Sarah Hug, Paul Salvador Inventado, Ayaan M. Kazerouni, Jane Lehr, Kanika Sood, and Zoë Wood. 2025. Reflecting on Practices to Integrate Socially Responsible Computing in Introductory Computer Science Courses. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701962

1 Introduction

Socially Responsible Computing (SRC) is a vital component of Computer Science (CS) courses, as it furthers educators' goals for advancing student learning outcomes and social justice through education. Ethics are an essential component of computing programs, and there is an identified need to suffuse ethics throughout the CS curriculum [10]. Incorporating SRC assignments into core courses that historically focus only on technical content is a pathway for achieving this. In addition to these strategic goals, incorporating SRC furthers practical considerations of student learning achievement, retention in academic programs, advancing diversity in computing programs and the computing workforce, and expanding the computing labor pool to meet industrial needs.

Incorporating SRC into CS courses is not automatic or easy. Instructors must identify or create materials or activities, make time in a course schedule, and incorporate SRC content into the course structure of content delivery, formative assessment, and summative assessment. Formal instruction in pedagogy and instructional design is uncommon for CS instructors, and that instruction rarely addresses SRC. Instructors routinely reuse or adapt pre-existing materials from text publishers and peers, and these materials often do not include SRC elements. Some students may be skeptical of the value of SRC material. Likewise, departmental colleagues, personnel procedures, and accreditation requirements may disincentivize inclusion of SRC. Finally, we recognize that highlighting power dynamics in computing and making space for discussions of social justice can be daunting.

In 2022, a group of computer science faculty from six California State University campuses formed a Faculty Learning Community (FLC) with the aim of combating these obstacles. Utilizing collaborative iterative design and implementation processes, with support from two social scientists embedded in the design team and modeled from social design experimentation as research and development method, the aims of the FLC are to provide peer mentorship for instructors incorporating SRC activities into CS0, CS1, and CS2 courses. This work is part of a collaborative grant in which department chairs and instructors collaborate to develop, modify, and implement SRC lessons that interweave attention to ethical, social, and political issues to position students to reflect and take action individually and collaboratively to create a more just world. Our approach to SRC supports students to explore computing in early CS courses with a communal goal orientation (in contrast to agentic/individualized), shown to improve achievement and retention for students with identities that are minoritized in CS.

In this study, we present initial results based on our co-creation and testing of SRC curriculum. In engaging in design and facilitating the FLC, the authors of this paper make pedagogical decisions about FLC content and direction, including the design of the monthly meetings, the summer face to face multi day workshop, the details and parameters of the collection of lesson plans for distribution across department teams, and participate in analysis and response to student-focused external evaluation efforts. The FLC involved frequent (monthly) virtual sessions in which participants presented methodologies, performed active learning activities, and mentored one another. These sessions culminated in annual, intensive, two-day in-person workshops. This structure forwarded the goals of establishing a community of practice, disseminating instructional methods, and group problem-solving. FLC participants include grant personnel and other instructors invited to engage in this work, a large proportion of whom are lecturer faculty.

SRC lessons were implemented in 84 sections over the past three semesters. Approximately 30 lecturers and faculty members participated each semester, with 80% retention from semester to semester. Over the course of three academic terms, spring 23 through summer 2024, in general our FLC met four times per term to tackle integrating SRC assignments into traditional computing courses (with a focus on courses in the first two years of computing majors). The FLC meetings were lead by a team of computing instructors and social scientists. FLC meetings were held over Zoom with one in person meeting per year and included computing instructors from at least six different primarily undergraduate institutions with a typical FLC meeting attendance including 15-20 total participants.

The FLC process surfaced a collection of promising practices for incorporating SRC into introductory CS courses. Practices were developed via an iterative co-design process as an approach to specific obstacles we feared and experienced in this implementation. These practices represent the culmination of the combined experiences of the FLC to date. In addition to understanding our own experiences as data that generates knowledge, initial evidence for the success of the FLC and support for the promise of these practices is multi-fold: 1) multi-site assignment adoption, including the expansion of implementers from grant PIs/Co-PIs to include newly recruited instructors and insights generated; 2) assignment innovation, including by recruited instructors (non-grant personnel); and 3) after the practices were codified during the 2023-24 academic year, they were presented to a fresh set of instructors in Summer 2024, and their feedback was solicited. We found that the practices were effective in addressing the concerns and obstacles identified by those new instructors. This suggests that the practices are generalizable: they are effective not just for the community that initially developed them, but for CS instructors in general.

In our FLC, we find that often computing instructors' fears about SRC implementation are larger than actual experiences in practice. While not the focus of this paper, earlier research has shown that by integrating a synergistic SRC context with required technical content we have been able to show no loss in technical achievements in later classes [23]. Student impact is also measured through external evaluation, and work under review. Briefly, the results indicate the SRC lessons improve student self-reported learning gains in technical content, self-reported interest in lessons, and self-reported engagement in social justice considerations of computing.

2 Literature Review

A growing body of scholarship advocates incorporating ethics and social responsibility in computing education [7, 9, 10, 30, 35]. This work is critical given reports of a decline in civic responsibilities among CS students in colleges [29]. At the same time, Horton et al. [22] report that even in their early work experiences like internships or research projects, students have reported facing issues relating to ethics and technology.

Multiple approaches to incorporating ethics and social responsibility into computing coursework have been proposed. For example, Cohen et al. [10] described a model in which participating undergraduate and graduate courses included multiple instances of responsible computing assignments, at least one of which counted toward the final course grade. Peck [30] has proposed a series of "Ethical CS" assignments for early courses that demonstrate to students that their choices as technologists can have impacts on people, for example, in automating decisions for housing allocations or hiring. However, challenges remain in introducing ethics in computing courses, such as a lack of faculty preparation and expertise [15, 32], resource constraints [8, 15], and even resistance [26, 32].

Along with its value in preparing students for civic participation, SRC with its focus on communal goals is valuable in broadening participation in computing. Our goal is to improve student retention in introductory computing courses. We aim to do this by bolstering their sense of belonging [19] in computing through coursework that explicitly demonstrates ways in which computing can impact society. This is especially important for students from underrepresented demographic groups. Numerous studies suggest that Hispanic/Latino, Asian, and Black students are less likely to report a high sense of belonging in computing than white students, and women are less likely to report a high sense of belonging in computing than men [4, 14, 18, 24, 28]. Previous research has also suggested that Hispanic/Latino, Asian, and Black students are more drawn to communal goals than white students, and women are more drawn to communal goals than men [12, 25]. Helping students see how their computing skills can support their goals and benefit society has the potential to strengthen their sense of belonging in computing [5, 6, 24], which in turn could improve their academic outcomes like performance, motivation, and persistence. Such focus on communal goals can be seen in research-based curricular innovations aimed at promoting diversity and inclusion in computing, for example, in NCWIT's Engagement Practices Framework which recommends educators to "help all students connect to computing by connecting computing to their students' lives" [16].

Scholars have advocated the need to go beyond ethics and support students in reflecting on their responsibility as future developers [10, 17, 31]. They position SRC as an approach that builds on ethical understanding towards responsible conduct, acknowledging the power that computing systems hold in society [17]. For example, Kazerouni et al. [23] present a data-centric, constructionist approach to introductory computing where students engage with personally or societally meaningful data, eventually building a website that reflects their analysis of societal impact. They report that the SRC-embedded CS0 course helped in promoting engagement in the class, and that the students who were in their course (treatment) performed better in the subsequent course compared to the students from the control course [23]. Gautam et al. [17] present their efforts of integrating socially responsible computing in an introductory course. Building on a series of socially-situated assignments and projects, they present students' reflections and values of power and responsibility in building computing systems. Reflecting on their experience, they list seven challenges they faced in integrating SRC, including the importance for instructors to be vulnerable in engaging the classroom discussions and ensuring that students reflect on the limits of technological solutions [17]. Cohen et al. [10] also draw attention to the importance of preparing educators to introduce socially responsible computing. It is used as a method to co-create change in local socio-political environments, with past work emphasizing access to healthy food [34], equity in K-12 computer science education [35] and educational opportunity for Dual Language Learners [20].

3 Methodology

Data sources for this work include: curriculum modules and lessons at each campus, the corpus of planning documents, recordings of all planning sessions, recordings of reflection meetings held with facilitators following each monthly meeting, student work samples, and aggregated student-level data from socially responsible computing course assignments. Data collection and analysis inform iterative social design experimentation [20, 21], leading to the next steps in the design and implementation of curricular units that engage students in socially responsible computing. The results presented in this paper document the shared meaning-making the team of FLC organizers have developed implementing variations of SRC lessons in classrooms across six campuses — presented as promising practices — and emphasize elements of lesson design and implementation based on social science research, active implementation and adjustment in classroom settings, and reflection in action and reflection on action [33]. We share how these promising practices guide our efforts; how they can address challenges and concerns for new and continuing SRC implementers; and the ways in which we have and will continue to test and co-design this approach.

4 The Practices

4.1 Choose the Domain Mindfully

As faculty at different stages of incorporating socially responsible computing into computer science college coursework, we have found that the selection of the social domain to integrate with the technical takes thoughtful consideration. Selection of a social topic that is relevant to student lives is imperative to enhance student engagement, and as such, instructors must consider institutional contexts, as well as student generational and cultural assets. For example, an SRC assignment focused on home mortgage calculations is not relevant to the majority of our students, who are low-income and often struggling to meet basic needs. In contrast, SRC assignments focused on on-campus housing assignments or access to low-income housing in the community may highlight significant student prior expertise and interest.

It is also important to select topics and issues that we, as instructors, feel comfortable facilitating, even when conversations become charged. Our experience indicates that instructors fear that domains that invoke strong opinions based on students' lived experiences can lead to conflicts—student contestations with one another and with the instructor's values. Preparing for possible conflict in class discussions is essential for successful SRC lesson implementation.

This dichotomy arose from the experiences of FLC faculty in selecting domains for SRC assignments. While CS instructors were eager to select topics that were relevant to students, they felt underqualified to facilitate discussions on the most politically contentious issues of the day. Faculty did feel equipped to incorporate issues including access to affordable housing, restaurant tipping, blood donation, bank regulation, and student personal finance.

One way that SRC faculty have managed conflict as part of choosing the domain mindfully is by structuring a collective knowledge base through assigned readings as part of the larger module. Directing how social domains are discussed creates a shared language for students to use and guides dialogue with a set of concepts that can focus on social justice discussions. When conflicts arise, discussion can be redirected towards elements of the readings. Even with instructor preparation and mindful domain choice, an SRC instructor may still be taken by surprise – by conflict or even by the direction that students take the assignment. We believe that SRC can be an opportunity for modeling vulnerability and improvisation in the classroom. For example, instructors in our team have modeled techniques such as saying "I don't know the answer, let's find out together," and providing classroom time and space at a later date for a deeper discussion of contentious and/or surprising topics. Faculty and instructors explored new lesson domains in the extended face-to-face summer workshop in 2024. Through coaching and faculty peer discussion a new instructor began the development of SRC material using a food security assistance program that made use of a web tool new to the college system. The larger team supported consideration of this domain, of which students may have some knowledge and experience, and brought out considerations for use in the lesson, such as issues of student eligibility for assistance.

4.2 Design For Synergy With Technical Material

As computing instructors, we do not want to lose focus on students' learning technical content. Designing socially responsible computing assignments that have synergy with the computing material is essential for student learning. When this link is less apparent, assignments can take longer than planned or be incomplete. Lack of synergy can also "other" SRC (discussed below). Social and ethical considerations must not be treated as separate or peripheral, but rather as an integral part of the technical learning process. The connection between technical learning outcomes and the social context in which we are situating the curriculum must be clear. This connection often involves discussing and identifying the overall solution, application, or algorithm to address the specific SRC topic and explicitly aligning the specific technical skills required for that solution.

For example, for courses focused on introductory programming, this can mean presentations on *data modeling* choices, *numerical weighting computations* given specific data representations, and the use of *conditionals* for decision making. Technical topics such as *filtering lists*, *looping mechanisms*, and *priority queues*, lend themselves to problems that involve decision-making and power relationships. Broader technical topics such as *security and privacy* or *efficiency and sustainability* can be explored in a wide variety of topics from a more design or analysis perspective.

In our FLC, it became clear that the assignments with a strong mapping between technical learning objectives and SRC topics were the most popular, reused, or commented on during FLC sessions. Examples include modeling student finances or lottery programs with object-oriented classes; simulating business procedures for blood donation, restaurant tipping, and bank runs; and comparing representation among graphics mesh datasets. These assignments demonstrated strong sociotechnical synergy and were aligned with appropriate technical skills for the given course. In some cases, this also means doing work to force better alignment. For example, for SRC assignments that require complex community data but the focus is on a more introductory computing concept [23], we observed that CS0 students were empowered to tackle the SRC topic of their choice when they were provided with an easy-to-use tool to 'flatten' any hierarchical dataset into a data model they were more familiar with processing. Finally, making sure that SRC assignments have synergy with technical learning objectives also addresses another often voiced concern by computing instructors that they do not have time to cover socially responsible computing.

4.3 Scaffold for Inclusivity

We aim to enhance students' sense of belonging, and in turn improve student success and retention rates. We focus on assisting educators in providing an inclusive learning environment to support all their students no matter their individual needs, backgrounds, experiences, and learning paces. By incorporating cultural diversity into the course material, they can support every student and allow them to connect and share. This practice is reflected in the selection of examples, exercises, case studies, and discussions such that all the students feel prepared to participate. Instructors should avoid using scenarios or references that may exclude or alienate certain students. For example, instead of focusing on algorithms relating to international travel, which presupposes wealth, concentrate on more universal domains such as approving credit card applications, displaying air quality information, blood donation, or beach clean-up. We encourage instructors to be culturally responsive and to recognize and utilize the community's cultural wealth and knowledge that students bring to the classroom [36].

Research shows that analyzing the themes and domains used in the classrooms by various institutions enabled more discussion among students and showcased a better understanding of the material [13]. The debate was prominent through the Canvas discussion board used for multiple assignments, such as the "bank run" and "tipping strategies," among other assignments used by multiple PIs and FLC members through different discussion platforms. Undergraduate classes, particularly the CS0 and CS1 courses, tend to be the primary platform for students transitioning from high school to a university setting. Their experience and performance in these introductory classes determine their career path and contribute significantly to the retention rate. Structuring the courses to draw upon the student's experiences can lay the foundation for student engagement and improve performance. Two ways to enable inclusivity in classrooms include:

1. **Select classroom material deemed fit for all.** Enhancing class material to introduce inclusivity and diversity and incorporating culturally responsive pedagogy can benefit students for a better student experience and boost performance.
2. **Limit the assumptions.** No underlying assumption of prior knowledge ensures the necessary context.

For example, an assignment on "Restaurant tipping strategies" was incorporated at one of the campuses for a CS0 course. Before the assignment, the students were given a pre-reading on the topic, which included key terminology, some sample ways (with examples) to split tips between employees, and additional pointers towards tipping techniques. This project is a practical example of how algorithms have political power, and the students are induced to think about "fairness" when distributing tips among workers. FLC faculty created the pre-reading assignment because they anticipated that some students—for example, international students—may not be familiar with the conventions of customers giving tips and employees pooling and distributing them, as tipping norms vary considerably around the world. Scaffolding this knowledge proved to be worthwhile, even for material that might seem rudimentary to instructors.

4.4 Structure With a Framework

Students say varied and surprising things when you open a discussion. Sometimes, it may make sense to constrain a discussion by asking students to "think or design from" an explicit framework/point of view. This allows you to ask, "how does that demonstrate x point of view?" / "how is that supported by or what is important according to y framework?" For example, if you are using a framework that prioritizes anti-racism and a student says something that might be experienced (by you or others) as racist during class discussion, you don't have to say "I think that is racist and this is why." Instead, you can say, "what would the creators of the anti-racist framework we are utilizing say about that statement? Is it something that we should include or discard in the context of this project since we are using that framework as a structure?" These frameworks can help guide students' thinking, provide a common language for analysis, and ensure a systematic approach to exploring SRC issues. In addition, they can be adapted and tailored to fit the specific needs and goals of your course and student population. We have found in the FLC that identifying and using frameworks helps computing instructors—who are often domain experts with technical materials but have hesitance or lack of experience managing classroom activities that include social contexts—to feel more comfortable and able to handle unexpected student reactions.

We have explored a number of frameworks in our work together. A number of our assignments (following Peck [30]) utilize a **power-to (capabilities) vs power-over (dominance or influence) distinction** [1], encouraging students to practice thinking and designing with a commitment to increasing rather than controlling human capacities. Some of us have also utilized an **"equity cognitive" frame** (in contrast to a "deficit cognitive structure"), where students are charged to focus not on "what do oppressed people lack that leads to their oppression?" but instead on the historical and structural factors that create and maintain systems of oppression [2]. A related framework is what Bensimon et al. [3] describe as **equity-mindedness**: an approach that is "color-conscious" rather than "color-blind"; recognizes "that beliefs, expectations, and practices assumed to be neutral can have outcomes that are racially disadvantageous"; takes institutional "responsibility for the elimination of inequality"; and is "[a]ware that while racism is not always overt, racialized patterns nevertheless permeate policies and practices in higher education institutions." Another productive framework is **design justice**, "an approach to design that is led by marginalized communities and that aims explicitly to challenge, rather than reproduce, structural inequalities" [11].

For example, one instructor, when having students design criteria for creating wait-list ordering for access to public housing, needed to assist the class in considering equity-mindedness and provide background reading when some students strongly advocated for including incarceration history as blocking criteria for housing. Students were allowed final design choices and were asked to submit reflections on their choices with respect to an equity-mindedness framework. As another example, the above-mentioned restaurant tipping assignment asks students to analyze tip allocation policies through the power-to versus power-over framework. These open-ended questions stimulated critical thinking and alternative viewpoints about social issues while bounding the level of contention within the comfort zone of students and instructors.

Structuring class discussions with frameworks is not a new idea in CS. Indeed, there is a long history of establishing a methodology in order to keep comparative analyses focused. When analyzing the performance of data structures, we establish the framework of big-O notation, acknowledging that it is not the only valid perspective on efficiency, but nonetheless we use asymptotic analysis to structure comparisons between arrays and linked lists. When analyzing the security of cryptographic systems we establish the framework of threat models and adversarial analysis, and likewise use that structure to keep class discussions focused while acknowledging that it is not the only valid perspective. In the same way, the above frameworks can constrain discussions of SRC issues within the bounds of a safe learning environment, aiding in facilitation.

4.5 Avoid Othering SRC Elements

In general, our FLC added new SRC assignments to courses that did not previously have SRC content. Naturally, in some cases, these new assignments stood out from the pre-established course content. Unlike pre-established content, SRC assignments were varyingly not on the syllabus, not mapped to a stated course learning outcome, not announced ahead of time, not part of assigned readings, not covered by examinations, graded differently than other assignments (e.g. graded for effort whereas all other assignments are graded for correctness) or not graded at all, depending upon the course. In some cases the instructional methods used for the SRC content were different from all other content, e.g. all class material was conveyed through slide presentations with the sole exception of SRC material that was structured as a group discussion.

These differences had the effect of *othering* SRC material in the eyes of our students. Othering SRC means signaling to students that SRC learning outcomes are categorically different from, or apart from, the technical learning outcomes of a course. Students are attuned to these sorts of signals and factor them into decisions about what to prioritize and value judgments about the comparative importance of course topics. This othering may have the negative effect of undermining the inclusion of SRC activities.

Consequently, it is a promising practice to avoid othering SRC material. SRC should be a first-class part of introductory CS courses. Like any other material, it should be: on the syllabus, linked to an explicit learning outcome, part of the routine course schedule, incorporated into the same formative and summative assessment regime as other material (i.e. reading assignments, quizzes, homework, examinations), and graded and presented in the same manner as other material. This necessitates planning for SRC inclusion at the time of designing a course, or writing a syllabus, well in advance of performing the SRC activities.

In our FLC, some instructors shared that they routinely othered SRC assignments due to honest oversight while others modified their entire course to focus on SRC assignments. Instructors said that it did not occur to them that they had othered SRC, or that this was problematic, until the group discussed this issue explicitly. The tendency of instructors to other SRC and the promising practice of avoiding that othering was an organic discovery that is straightforward in hindsight. The promising practice of avoiding this othering, and planning ahead, was recognized as an insight from the mixed implementations and sharing experiences. To address implicit othering that some instructors recognized might be undermining their FLC efforts, instructors worked on addressing this for example by adding SRC to the syllabus topic list or adding an exam question.

4.6 Reuse And Build On Existing Resources

In our experience, we found that FLC instructors struggled to create new SRC assignments. They had several questions about how to choose the domain, incorporate SRC into class activities without needing more time, and SRC assignments' effectiveness in teaching a CS concept. Such experiences are intimidating and can result in faculty losing interest in SRC assignments.

It is a promising practice to reuse existing SRC assignments. You can leverage existing assignments, case studies, and topics that have been successfully used to incorporate SRC, and adapt and customize these resources to fit the specific needs and interests of your students and the goals of your course. Our team has compiled several SRC assignments that are publicly available to facilitate this adoption process[1]. Several of our assignments have been reused at different institutions and were found to work well in terms of student familiarity and engagement.

Reusing assignments turned out to be effective, especially for instructors who are new to SRC. We have seen SRC assignments reused in three ways: **1) Reuse without modification. 2) Modifying the course content.** Modifications might be straightforward, such as translating a Python-based SRC assignment into C++. This might involve translating the instructions, questions, and sample code into the new language. In some cases, the theme can be reused for new technical content, such as reusing the tipping strategy theme for loops instead of conditions. **3) Adopting an SRC theme.** Some assignments already used in class may lend themselves to naturally adopting an SRC theme. For example, an assignment on file reading and writing might instead use SRC-related data. For example, instead of reading arbitrary data, you can use SRC-related data like CS enrollment by gender and race. You can reuse existing materials on access to education from existing SRC assignments. We found, having a community open to *re-mixing* themes and technical focus while sharing lessons learned to be very productive.

Reusing SRC assignments brings several benefits. First, it reduces the work required to create the assignment. You can reuse materials like introductory readings, examples, and references about the SRC domain. Second, it promotes replication. An SRC assignment might teach a certain group of students and engage them with the content effectively, but it might not work as well for a different group. Reusing assignments in different learning contexts and student populations helps evaluate the SRC assignment. We can be more confident sharing SRC assignments tested in multiple learning settings. Third, we learn more about the properties of an assignment that make it effective when successfully used in different learning settings. For example, we discovered that the "tipping strategy" assignment was effective because some students at institutions that used the SRC assignment worked in the service industry. Therefore, it is important to choose domains that your students find relatable. Fourth, it encourages research when an assignment does not work as effectively as in other learning settings. We can investigate why an SRC assignment's effectiveness differs between institutions and encourage identifying ways to refine the assignment or find a different, more suitable SRC assignment. Finally, the experience we gain from replicating SRC assignments helps us gain insights to develop promising practices for creating new SRC assignments.

Examples of reuse from our FLC include one of the grant PIs reusing an assignment based on existing SRC work using the power-over and power-to framework. The assignment used tipping calculation as the assignment domain, requiring students to practice using arithmetic operations, loops, and conditional statements in Java for a CS0 course [17]. One group of FLC participants modified the assignment to use the same pre-reading material, but introduced POGIL-like [27] worksheets for an in-class activity, assigned a discussion board for student reflections, and used C++ for a CS0 course. Another participant modified the assignment into a take-home assignment using Python for a teacher credentialing course.

[1] https://curriculum.bpcsrc.org

5 Discussion

In this paper, we draw from three datasets: our own experiences and collective knowledge production developing, implementing and revising SRC curricula for early CS courses; insight from others in their participation in the FLC and processes of assignment implementation; and, our observations of the ways in which these practices serve as a resource in the generation of new assignments by new participants in the FLC. In this work, we employed a social design experimentation methodology to describe the emergence, refinement, and testing (to date) of six promising practices for the implementation of socially responsible computing curricula.

In situ, these practices are integrated to provide support, rather than utilized "practice-by-practice" as presented here. For example, in the Summer 2024 FLC workshop, based on the presentation and active engagement with these practices, participants who were teaching data structures courses (for which we had no curriculum at the time), were able to ideate for using priority-queues with power related SRC contexts such as human ranking/prioritization policies and likewise for tying asymptotic efficiency to energy and sustainability concerns in computing. Further evidence of the value of this work and these practices is that for the most recent in person gathering for the FLC, four 'newer' faculty (of six presentations) presented their work on SRC assignments, including completely novel work on examining injustice in 911 response times.

Socially Responsible Computing (SRC) education synergistically integrates attention to ethical, social, and political issues in CS education and is a powerful tool designed to enable students to reflect and take action individually and collaboratively to create a more just world. In addition, research suggests that shifting from agentic/individualized to communal goal orientation has positive impacts for students minoritized in CS. Incorporating SRC into CS courses is not automatic or easy. Faculty (realistically and reasonably) require support to do this transformative work. We have described our approach to this task of professional development via a community of practice. In addition to suggesting directions for the development of SRC curricula in early CS courses, our use of social design experimentation as a framework and the collaborative, mutual partnership between CS faculty and social science team members has been an impactful experience for all participants. As a community, we regularly reflect and celebrate the impact of the FLC on our ongoing development as CS instructors. Even during finals week, we are inspired by the 15-20 instructors who show up for FLC meetings ready to talk about new assignments they are considering or reflect on how their recent work has gone.

Acknowledgments

This work was supported by National Science Foundation awards 2216687, 2216672 2216513, 2216672, 2216520, and 2216575.

References

[1] Amy Allen. 2022. Feminist Perspectives on Power. In *The Stanford Encyclopedia of Philosophy* (Fall 2022 ed.), Edward N. Zalta and Uri Nodelman (Eds.). Metaphysics Research Lab, Stanford University.

[2] Estela Mara Bensimon. 2005. Closing the achievement gap in higher education: An organizational learning perspective. *New Directions for Higher Education* 2005, 131 (2005), 99–111. https://doi.org/10.1002/he.190 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/he.190

[3] Estela Mara Bensimon, Alicia C Dowd, and Keith Witham. 2016. Five principles for enacting equity by design. *Diversity and Democracy* 19, 1 (2016), 1–8.

[4] Jennifer M. Blaney and Jane G. Stout. 2017. Examining the Relationship Between Introductory Computing Course Experiences, Self-Efficacy, and Belonging Among First-Generation College Women. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 69–74. https://doi.org/10.1145/3017680.3017751

[5] Kathryn L Boucher, Melissa A Fuesting, Amanda B Diekman, and Mary C Murphy. 2017. Can I work with and help others in this field? How communal goals influence interest and participation in STEM fields. *Frontiers in psychology* 8 (2017), 901.

[6] Bo Brinkman and Amanda Diekman. 2016. Applying the communal goal congruity perspective to enhance diversity and inclusion in undergraduate computing degrees. In *Proceedings of the 47th ACM technical symposium on computing science education*. 102–107.

[7] Noelle Brown, Koriann South, Suresh Venkatasubramanian, and Eliane S. Wiese. 2023. Designing Ethically-Integrated Assignments: It's Harder Than it Looks. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 177–190. https://doi.org/10.1145/3568813.3600126

[8] Noelle Brown, Koriann South, and Eliane S Wiese. 2022. The Shortest Path to Ethics in AI: An Integrated Assignment Where Human Concerns Guide Technical Decisions. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*. 344–355.

[9] Francisco Castro, Sahitya Raipura, Heather Conboy, Peter Haas, Leon Osterweil, and Ivon Arroyo. 2023. Piloting an Interactive Ethics and Responsible Computing Learning Environment in Undergraduate CS Courses. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 659–665.

[10] 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*.

[11] Sasha Costanza-Chock. 2020. *Design justice: Community-led practices to build the worlds we need*. The MIT Press.

[12] Amanda B Diekman, Mia Steinberg, Elizabeth R Brown, Aimee L Belanger, and Emily K Clark. 2017. A goal congruity model of role entry, engagement, and exit: Understanding communal goal processes in STEM gender gaps. *Personality and social psychology review* 21, 2 (2017), 142–175.

[13] John Dunlosky, Katherine A Rawson, Elizabeth J Marsh, Mitchell J Nathan, and Daniel T Willingham. 2013. Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public interest* 14, 1 (2013), 4–58.

[14] Mica Estrada, Myra Burnett, Andrew G. Campbell, Patricia B. Campbell, Wilfred F. Denetclaw, Carlos G. Gutiérrez, Sylvia Hurtado, Gilbert H. John, John Matsui, Richard McGee, Camellia Moses Okpodu, T. Joan Robinson, Michael F. Summers, Maggie Werner-Washburne, and MariaElena Zavala. 2016. Improving Underrepresented Minority Student Persistence in STEM. *CBE—Life Sciences Education* 15, 3 (2016), es5. https://doi.org/10.1187/cbe.16-01-0038 PMID: 27543633.

[15] Casey Fiesler, Mikhaila Friske, Natalie Garrett, Felix Muzny, Jessie J Smith, and Jason Zietz. 2021. Integrating ethics into introductory programming classes. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*.

[16] The National Center for Women & Information Technology (NCWIT). [n.d.]. Community Inquiry. https://ncwit.org/resources/engagement-practices-framework/. Accessed: July 17, 2024.

[17] Aakash Gautam, Anagha Kulkarni, Sarah Hug, Jane Lehr, and Ilmi Yoon. 2024. Socially Responsible Computing in an Introductory Course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 373–379.

[18] Catherine Good, Aneeta Rattan, and Carol S. Dweck. 2012. Why do women opt out? Sense of belonging and women's representation in mathematics. *Journal of Personality and Social Psychology* 102, 4 (2012), 700–717. https://doi.org/10.1037/a0026659

[19] Carol Goodenow. 1993. The psychological sense of school membership among adolescents: Scale development and educational correlates. *Psychology in the Schools* 30, 1 (1993), 79–90. https://doi.org/10.1002/1520-6807(199301)30:1<79::AID-PITS2310300113>3.0.CO;2-X

[20] Kris D Gutiérrez and A Susan Jurow. 2018. Social design experiments: Toward equity by design. In *Cultural-Historical Activity Theory Approaches to Design-Based Research*. Routledge, 79–112.

[21] Kris D Gutiérrez, A Susan Jurow, and Sepehr Vakil. 2020. A Utopian Methodology for Understanding New Possibilities for Learning. *Handbook of the cultural foundations of learning* (2020), 330.

[22] Diane Horton, David Liu, Sheila A. McIlraith, Steven Coyne, and Nina Wang. 2024. Do Embedded Ethics Modules Have Impact Beyond the Classroom?. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 533–539. https://doi.org/10.1145/3626252.3630834

[23] Ayaan M. Kazerouni, Jane Lehr, and Zoë Wood. 2024. Community Action Computing: A Data-centric CS0 Course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 646–652. https://doi.org/10.1145/3626252.3630807

[24] Colleen Lewis, Paul Bruno, Jonathan Raygoza, and Julia Wang. 2019. Alignment of Goals and Perceptions of Computing Predicts Students' Sense of Belonging in Computing. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. ACM, Toronto ON Canada, 11–19. https://doi.org/10.1145/3291279.3339426

[25] Colleen Lewis, Paul Bruno, Jonathan Raygoza, and Julia Wang. 2019. Alignment of goals and perceptions of computing predicts students' sense of belonging in computing. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. 11–19.

[26] C Dianne Martin and Elaine Yale Weltz. 1999. From awareness to action: Integrating ethics and social responsibility into the computer science curriculum. *ACM Sigcas Computers and Society* 29, 2 (1999), 6–14.

[27] Chris Mayfield, Sukanya Kannan Moudgalya, Aman Yadav, Clif Kussmaul, and Helen H. Hu. 2022. POGIL in CS1: Evidence for Student Learning and Belonging. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 439–445. https://doi.org/10.1145/3478431.3499296

[28] An Nguyen and Colleen M. Lewis. 2020. Competitive Enrollment Policies in Computing Departments Negatively Predict First-Year Students' Sense of Belonging, Self-Efficacy, and Perception of Department. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 685–691. https://doi.org/10.1145/3328778.3366805

[29] AM Núñez, MJ Mayhew, M Shaheen, and LS Dahl. 2021. Let's teach computer science majors to be good citizens. *The whole world depends on it. Edsurge* (2021).

[30] Evan Peck. 2017. The Ethical Engine: Integrating Ethical Design into Intro Computer Science. *blog, Bucknell HCI* 5 (2017).

[31] Jean J Ryoo, Alicia Morris, and Jane Margolis. 2021. "What happens to the Raspado man in a cash-free society?": Teaching and learning socially responsible computing. *ACM Transactions on Computing Education (TOCE)* 21, 4 (2021), 1–28.

[32] 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 (TOCE)* 19, 4 (2019).

[33] Donald A Schön. 2017. *The reflective practitioner: How professionals think in action*. Routledge.

[34] Leah Anne Teeters and A Susan Jurow. 2019. Generating equity-oriented partnerships: A framework for reflection and practice. *Journal of Community Engagement and Scholarship* 11, 1 (2019), 27–37.

[35] Sepehr Vakil. 2018. Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard educational review* 88, 1 (2018), 26–52.

[36] Tara J Yosso. 2005. Whose culture has capital? A critical race theory discussion of community cultural wealth. *Race ethnicity and education* 8, 1 (2005), 69–91.