Integrating Ethics into Introductory Programming Classes

Casey Fiesler
casey.fiesler@colorado.edu
University of Colorado Boulder

Mikhaila Friske
mikhaila.friske@colorado.edu
University of Colorado Boulder

Natalie Garrett
natalie.garrett@colorado.edu
University of Colorado Boulder

Felix Muzny
f.muzny@northeastern.edu
Northeastern University

Jessie J. Smith
Jessie.Smith-1@colorado.edu
University of Colorado Boulder

Jason Zietz
jason.zietz@colorado.edu
University of Colorado Boulder

ABSTRACT

Increasing attention to the role of ethical consideration in computing has led to calls for greater integration of this critical topic into technical classes rather than siloed in standalone computing ethics classes. The motivation for such integration is not only to support in-situ learning, but also to emphasize to students that ethical consideration is inherently part of the technical practice of computing. We propose that the logical place to begin emphasizing ethics is on day one of computing education: in introductory programming classes. This paper presents one approach to ethics integration into such classes: assignments that teach basic programming concepts (e.g., conditionals or iteration) but are contextualized with real-world ethical dilemmas or concepts. We report on experiences with this approach in multiple introductory programming courses, including details about select assignments, insights from instructors and teaching assistants, and results from surveys of a subset of students who took these courses. Based on these experiences we provide preliminary plans for future work, along with a roadmap for instructors to emulate our approach and suggestions for overcoming challenges they might face.

CCS CONCEPTS

• **Social and professional topics** → *Model curricula.*

KEYWORDS

ethics, introductory programming, CS1, social impact, assignments, university, undergraduate, content

ACM Reference Format:
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 (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432510

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

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432510

1 INTRODUCTION

> If, as we claim, social and ethical concerns are an integral part of designing computer systems, then why are they frequently absent in computer science education?

This question, posed in a 1994 *Communications of the ACM* article [9], is in many ways as relevant now as it was 25 years ago. Though interest and coverage of computing ethics in higher education is increasing [8, 29], so is public and academic discourse around harms caused by technology and responsibility held by technologists [10, 31]. Therefore, it is important that we consider not just *whether* social and ethical concerns are present in CS education, but *how*.

In the United States, the Accreditation Board for Engineering and Technology (ABET) altered criteria for computer science programs in the mid-2000s to include providing students with "an understanding of professional, ethical, legal, security and social issues and responsibilities" and "an ability to analyze the local and global impact of computing on individuals, organizations, and society" [14]. However, there are not clear guidelines for how to accomplish this, and different programs handle it differently. One analysis showed that the most common strategy for a sample of ABET-accredited CS programs was a required junior or senior level standalone ethics or professionalism class, either taught by the department or by a different department such as Philosophy [14].

Standalone ethics classes are important pedagogically and have been examined in prior work [8], which also suggests the benefits of standalone classes paired with curricular integration [19]. However, we see three potential flaws to standalone ethics classes alone as a solution, particularly those taught near the end of a degree program: (1) many students take computing courses, not just majors, and therefore many students learning computing might not have any exposure to ethics content; (2) teaching ethics as divorced from technical classes fails to emphasize that ethical consideration should be an actual part of that technical practice; and (3) students form values associated with their profession along the way, so if they spend 2 to 3 years learning computing without hearing about ethics, it may be too late by the time they do (similar to arguments around creating good code documentation habits from the start [4]).

We piloted an intervention to integrate ethics into introductory programming classes as a possible solution that tackles all three of these shortcomings. In addition to the benefits of ethics integration into technical classes generally, a solution for which we are seeing an increasing number of calls to action [8, 10, 11, 23, 24, 27, 30], targeting introductory programming classes has the additional benefits of reaching a large number of students who may not go on

to take more computing classes, as well as emphasizing to students that ethics is part of computer science from the very beginning.

This paper describes a pilot program that replaced existing assignments in these classes with assignments contextualized with ethical dilemmas and concepts. We report on the benefits and challenges of this approach based on instructor and teaching assistant observations and a post-class student survey. Based on these experiences we provide suggestions for emulating our approach and for overcoming challenges instructors might face.

2 RELATED WORK

In a letter to the editor of *Communications of the ACM* in 1996, a CS professor argued that ethical consideration is "not doing computer science" and that it was "difficult to imagine a computer scientist teaching these things" [18]. This opinion came in response to an NSF-funded report that recommended that in addition to standalone classes, ethics and social impact should be integrated into core CS classes [19]. Resistance to this kind of integration might come from that idea that the material is not appropriate for the CS classroom, from an instructor not knowing what to teach or how to teach it [8, 24], or from concerns about time being at a premium when there is so much technical content to cover [3]. However, not addressing potential ethical issues as they arise in-situ during computational learning risks marginalizing ethics or reinforcing the idea that it is for someone else to worry about rather than a necessary part of the daily practice of a technologist [8].

However, though integration is still not standard practice (with standalone classes more common [14]) even for topics for which ethics is highly relevant such as machine learning [10, 24], there are a number of strong examples. For example, the Embedded EthiCS program embeds philosophy graduate students and postdocs into courses throughout the computer science curriculum, modifying each course to include relevant ethics content [11]. Though this is a resource-intensive solution, it has been effective in overcoming the shortcomings of standalone classes and addressing one of the significant challenges of integration, reluctance of instructors to teach the material themselves [11]. Others have integrated ethics content into specific classes, including human-computer interaction [30], data science [24, 27], and even lower-level programming classes [3]. Overall, evaluations and experiences of these interventions suggest that they benefit students, though they often require significant course overhauls. For example, a CS2 intervention included new types of assignments with written reflections [3].

Another benefit of incorporating ethical thinking and speculation into programming classes is that ethical dilemmas provide real-world context. We know from prior work that incorporating context and concrete application domains can increase success rates and retention in introductory computing courses [13, 20]. Prior work has also shown that including socially relevant examples and an emphasis on the social impact of computing has a positive impact on participation and retention in computing for people from under-represented groups such as women [16, 26]. Additionally, ethical considerations permeate topics such as bias, workforce diversity, and accessibility, all important topics for cultivating culturally competent students who are able to think inclusively [17, 33].

Table 1: Overview of courses for integration, along with evaluation method and analysis.

Course	Semester(s)	Integration	Survey?	Quant?
CS-A	Fall 2019	2 assignments	No	No
CS-B	Spring 2020	2 assignments	Yes	Yes
INFO-A	Fall 2019/Spring 2020	2 assignments	Yes	No
INFO-B	Summer 2020	throughout	Yes	No

Though in general, contextualized assignments are not uncommon in introductory programming classes, ethics specifically still appears to be rare. A 2006 survey study showed that "ethics" was considered by CS1 instructors to be one of the least "important" topics to cover [25], and a 2019 analysis of CS1 syllabi showed no mention of ethics among learning outcomes for students [1]. However, towards a goal of emphasizing to students that ethics is essential to the role of a computing professional, introductory programming is the perfect place to begin ethics integration. Our intervention focused on replacing existing assignments with new assignments contextualized with ethical dilemmas and concepts.

3 COURSE DESCRIPTIONS

In the 2019/2020 academic year, we piloted our approach in three different introductory programming courses at University of Colorado Boulder, a large state university in the United States. Table 1 provides an overview of these courses, along with notes about evaluation methods (see Section 5).

CS-A. Primarily serving non-CS-majors in STEM and engineering as well as students who haven't yet declared a major, this course is taught in Python and had 170 students enrolled in Fall 2019. We replaced 2 of 6 assignments with the contextualized assignments described below. In addition to specific discussions related to these assignments in recitation sections, there was one guest lecture from an ethics expert partway through the semester and one lecture on AI bias from the instructor near the end of the course.

CS-B. Primarily serving computer science majors and engineering majors, this course is taught in C++ and had roughly 700 students enrolled in Spring 2020. We replaced 2 of 8 assignments, and following each assignment a guest speaker gave a 15 to 30 minute lecture and led discussion of the relevant topic.

INFO-A. Serving information science majors as well as non-majors (primarily from Strategic Communication, Communication, and Journalism), this class is taught in Python. 195 students were enrolled in Fall 2019 and 214 students were enrolled in Spring 2020. We replaced 2 of 8 assignments, and a guest speaker gave a 15 to 30 minute lecture following the assignments.

INFO-B. This course is the same as INFO-A, but was taught in Summer 2020 with 17 students. It was also redesigned, where based on the success of these previous classes, ethics was integrated throughout the entire semester. Five "case studies" (modules) each touched on specific ethics concepts as they related to computing, technology, and coding. Each module included both class discussion and a coding assignment or activity with a written reflection.

Though this paper is focused on our individual assignment-based intervention, we include some information about INFO-B in order

to reflect on and make suggestions for even more robust ethics integration. In describing evaluations (such as observation or student comments) we specifically note when we are referring only to INFO-B in order to separate out any measures of success. Also note that though INFO-B included graded assignments (e.g., reflections) related to ethics, the other courses with our primary intervention did not, largely for reasons discussed in Section 6.3.

4 INTERVENTION

Previous ethics interventions in programming classes (for example, CS2 and data structures [3]) typically have involved new material and types of assignments (e.g., written reflections). In considering options for integrating ethics into existing introductory programming courses, we ultimately decided to start with the approach that would result in the least amount of work for the instructors of those courses, who are often overburdened with very large classes and limited time for course development.

Ethics experts unaffiliated with the classes took existing assignments and altered them so that they taught the same technical content, but were re-contextualized to inherently bring up ethical concepts and current ethical dilemmas in tech. Instructors and teaching assistants then further refined them to ensure that they were in line with class goals and learning objectives, and formatted appropriately. They also created grading criteria.

For the three courses where we piloted this approach (CS-A, CS-B, and INFO-A), we replaced two assignments. INFO-B, taught in the summer following this intervention, experimented with additional assignments as well as further ethics integration with class discussion and reflection exercises.

We will describe in detail the two assignments from the first three courses as our primary intervention.[1] There were two versions of each assignment: one in Python and one in C++, depending on the language used in the class.

4.1 Personalized Ads

This assignment covered conditionals and booleans, with learning objectives that included implementing one-way, two-way, and multi-way decisions using if/else statements and understanding boolean expressions and data types. The following context was provided for the assignment:

> Your job is to write a program that decides on an ad to serve to a person on a social media platform. The personalized ad program will prompt the user for information and then return text that describes ads based on their inputs.

The assignment then walks the student through writing functions that are increasingly complex–both with respect to the type of decision and conditional structure, and to the type of personalization and data-based inference involved. Here are simple explanations for each:

(1) Advertise dog food to a social media user if they own a dog.
(2) Research has shown that algorithms can determine whether a social media user is likely to be an extrovert or an introvert based on the number of friends they have on the platform. Research has also shown that extroverts are more likely to be dog people and introverts are more likely to be cat people, and that people are more likely to click on ads for any product that remind them of their pet. Based on the number of friends that the user has, serve an ad that includes a dog or a cat.
(3) The platform has data about likely income averages based on zipcode and how incomes vary based on life stage (e.g., college students have less disposable income no matter where they live). Serve ads for expensive versus inexpensive products based on where the user lives and how old they are.

This assignment was designed to emphasize to students: (1) how their data might be used for advertising purposes; (2) how inferences about someone can be drawn from data, and therefore information can be known about someone beyond what they explicitly state; and (3) how what is known about someone can influence not only what is advertised to them, but how it is advertised. In discussion following the assignment (either in recitation sections or in lecture), the instructor or a guest speaker discussed two well-known examples: if Target's algorithms can identify whether a customer is pregnant [32]; and the Cambridge Analytica scandal [28]. The introvert/extrovert function specifically tees up a conversation about Cambridge Analytica, where the nuance is that based on Facebook data, the manipulation was not in determining what to advertise to a user, but rather that inferred personality traits (such as introversion versus extroversion) suggested *how* to advertise.

4.2 College Admissions Algorithms

This assignment covered lists and file input/output, with learning objectives related to iterating through a list and reading and writing data from and to files. The following context was provided:

> As you know, the college admissions process involves a lot of types of data from prospective students to make decisions. With the number of applicants increasing, colleges may begin relying on algorithms to select which applications should receive more intensive human review. An algorithm could use quantitative data–such as GPA and SAT score–to provide initial recommendations. In fact, there is more data available than ever. Many colleges even track data about prospective student engagement - e.g., whether they open emails, visit the college website, engage on social media, etc. [2] This creates a "demonstrated interest" value. Based on a recent survey of college admissions officers [15], we know some of the weights that humans tend to give to these different types of data. Your task will be to create a program that iterates through a list of data points and provides a recommendation for which prospective students are likely to be the best candidates for admission.

For the assignments, students were provided with a file of fictional prospective student data points organized in lists that included GPA, SAT score, a score of high school curriculum difficulty, and a "demonstrated interest" score. The assignment asked students to provide overall scores for each prospective student based on provided weights and then return a list of students recommended by

[1] Exact assignments and supplementary materials are available at www.internetruleslab.com/responsible-computing

the algorithm. The rest of the assignment walked them through additional transformations that explored how the algorithm might systematically miss certain kinds of edge cases. For example, what if a student has a 0 for demonstrated interest because they don't use social media or have access to a home computer? What if a student has a very high GPA but their SAT score is low enough to bring their score down; could this mean that they had a single bad test taking day? Students then wrote additional algorithms that checked for outliers in the data. Discussion following this assignment focused on algorithmic decision-making and representations of people in data, bias and fairness, human judgment in algorithmic decisions, and the appropriate role of algorithms in decision-making.

5 STUDENT SURVEYS

In Spring and Summer 2020 we conducted surveys at the end of class in CS-B, INFO-A, and INFO-B (CS-A was only taught in Fall), with approval from our university Institutional Review Board. We did not provide any compensation or extra credit, though did give students time in class to take the survey, which was administered by a researcher who was not an instructor or TA for the course. We suspect that partly due to disruptions from COVID-19, the response rate was quite low in Spring: 135 out of 700 students for CS-B and 7 out of 214 for INFO-A. For Summer's INFO-B, 11 out of 17 students responded. Because sample sizes were so low for the other classes, the quantitative data we report is only from the 135 students who took CS-B in Spring 2020. The qualitative findings reflect responses from all courses, though we separate out responses that come from the Summer course where the ethics integration was more robust, in order to clarify the impact of the initial assignment-based intervention. Table 1 provides an overview of these classes and these evaluations and analyses.

We asked optional demographic questions so we could appropriately describe the student population. For students in CS-B who answered these questions, the mean age was 19; 59 men, 43 women, 1 non-binary individual, and 1 who preferred not to disclose; 51 freshmen, 37 sophomores, 14 juniors, and 2 seniors; students were majority white (66%); and 40% were computer science and computer engineering majors; 33% other engineering majors, and the rest undecided or other majors, mostly STEM.

For the quantitative results described for CS-B in Table 2 and Figure 1, we asked questions that directly compared learning and engagement for our new assignments to the other assignments that were not contextualized with ethical issues. We operationalized engagement by how "interesting" an assignment was. These included the following quantitative questions:

(1) How much did you learn from the [personalized ads or admissions algorithm] assignment compared to your other assignments? ("less," "about the same," or "more")
(2) How interesting was the [personalized ads or admissions algorithm] assignment compared to your other assignments? ("less," "about the same," or "more")
(3) In general, how interested would you be to see more assignments that bring up ethical issues in your future programming classes? (on a 5-point scale of "very uninterested" to "very interested")

Table 2: Responses to Assignment-Based Questions

Assignment	Less	About the Same	More
Ads - Learning (Q1)	12%	75%	13%
Admissions - Learning (Q1)	10%	57%	33%
Ads - Engagement (Q2)	20%	57%	23%
Admissions - Engagement (Q2)	13%	51%	36%

With respect to comparing the ethics-based assignments to other assignments, the results were mixed but trend positive, suggesting that the assignments had value and, importantly, did not take away from the class. Very few students said they *learned* less from these assignments, and many said that they were *more interesting* and that they *learned more* (see Table 2).

The third question gives another measurement of engagement, by asking how interested students would be in this kind of content in their future programming classes; we found that the majority of students would be interested in such content (see Figure 1). 51% said that they would be somewhat or very interested, and only 15% said that they would be somewhat or very uninterested.

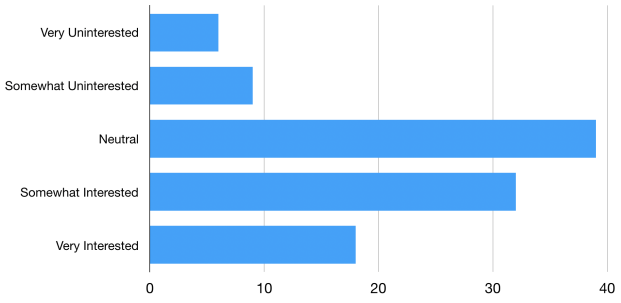

Figure 1: Chart of responses, in percentages, to level of interest in ethics-based assignments in more classes.

In analyzing free response answers to one open question ("Describe an ethics concept that came up in class this semester."), we found about equal mention of the issues from our intervention–e.g., biased algorithms, personalization, and privacy. Unexpectedly, we also noted that a number of students equated ethics with plagiarism, which was not part of our intervention. However, this suggests plagiarism might be an interesting topic for a more robust discussion around ethics that could tie into broader ethical issues than the honor code, such as real-world practice and professionalism.

Despite the fact that overall the responses suggested that students picked up on ethical concepts due to our intervention, it was still clear that not every student "got it." For example, one student claimed that for the admissions algorithm assignment, there were no ethical implications for the programmer because they were just doing what they were told. This response suggested ways that we can do a better job in framing very basic concepts like power, bias, and especially responsibility.

In INFO-B, the small, summer class, where ethical concepts were incorporated into every assignment, we asked free-response questions about both learning ("Which assignment in this class do you think most helped you learn a technical concept? What did you learn?" and "Which assignment in this class do you think most helped you learn an ethical or social concept? What did you learn?") and engagement ("Which assignment in this class was most interesting to you and why?"). There was the most engagement, in terms of both interest and ethical concepts, with an assignment about autonomous vehicles. For example, the students wrote about learning how there are often no "right" answers and how important ethical audits are for AI. Additionally, when describing technical learning some students also brought up ethical concepts–for example, "This assignment was very helpful in learning how to parse data, and I learned that code does not have empathy." Finally, as also noted with our own observations detailed next, students mentioned how interesting they found assignments that felt "real" and had real-world implications.

6 INSIGHTS FROM INSTRUCTORS

The authors of this paper include three instructors and two teaching assistants from these courses, as well as the guest speakers/ethics experts who contributed to the content and discussion. Based on our collective experiences, we lay out our observations of benefits to the students (also based on survey responses described above).

6.1 Observed Benefits for Students

Based on in-class discussion, the idea that ethical issues are important in computer science right now is not new for many students; sometimes they even bring up ethical issues unprompted. With just a small push to include the content in class, it actually does not require much work to help them make the leap from "ethics matters in tech" to "I should be thinking about this when writing code." Though more formal empirical research is needed to determine actual impacts in terms of learning and attitude shifts, our observation of students in class is that, though not all were deeply engaged, those that were engaged were very engaged.

Another benefit of assignments contextualized with real-world ethical dilemmas is that, as we know from prior work and saw reinforced here, students enjoy real-world contexts for assignments in general. We saw that students were especially engaged when they were thinking about tangible ethical issues where they could see the direct application, such as bias in algorithms. Students expressed in class that they appreciated that they could apply this content to current events, the tech they use every day, and their personal data.

We also saw that students enjoyed the creative components; for example, in the personalized ads assignment, they had the freedom to write their own text for the ads, and often did more work there than expected. In general, even assignments for beginning programming concepts typically have *some* kind of context; the ones we chose were engaging because they covered topics that the students were already familiar with and knew were important. We of course cannot entirely disentangle here the effect of real-world context versus ethic specifically, though this would be an important question for future work.

6.2 Observed Challenges for Students

The most significant barrier we noticed to student engagement was that when students struggle with the technical material, they seem less engaged in the ethical components. Instead, they are focused on trying to understand the part they are being graded on. We suspect this is why engagement with ethical concepts seemed to be the lowest in the Fall and Spring iterations of INFO-A, which included a large number of students from non-technical majors, students who in general struggle the most with class content. They also showed less creativity in their assignments when given the opportunity, possibly because they were more worried about right answers and/or were spending more time on the technical aspects.

6.3 Challenges for Instructors

One goal with this intervention was to develop an approach that would be very little work for the instructors–either by the development of shared resources (such as the assignments we have created) or by working with outside experts to develop new assignments. We chose to replace assignments, as this typically does not require finding extra time in class, changing grading schemes, or adding a lot of new content.

Although one design constraint of these assignments was to minimize the amount of additional work for the instructor, translating assignments between classrooms presented challenges such as addressing specific learning goals or standardizing formatting. An assignment created for one class cannot necessarily be used "out of the box" in another class, particularly if the new class covers less material than the original class. We came to realize that the process would be much smoother if the instructors themselves were leading the adaptation, with assistance from ethics experts, if necessary. This is also a task that might be well-suited for teaching assistants.

Another challenge is that large classes often rely on auto-grading, which makes it difficult to (1) increase student agency (e.g., making outside-the-box decisions or being creative); and (2) evaluate ethical learning. Because of resource constraints and/or consistency in grading, it can be difficult to add hand-graded content such as open-ended questions about ethical concepts. For our interventions, we simply included this material without grading it—but it is possible that the students would be more engaged or more convinced of the importance of the material if it were "on the test."

Finally, adding ethics-related content to class discussion requires training for instructors and teaching assistants, or guest lectures from experts. For our intervention, we largely relied on the latter, but this is not necessarily sustainable and can be improved upon.

7 DISCUSSION AND RECOMMENDATIONS

Based on our experiences with these interventions, here we make an argument for why this approach would be beneficial to others, and then suggest methods of implementation and possible solutions to the challenges we described above.

One concern for adding this content to classes might be that it would take away from the important technical content of the class. However, based on survey results and instructor observation, we have no reason to think that our intervention detracted from technical learning. In fact, one student, who had taken a different coding course before the INFO-B course, told the instructor that they

learned more from the class than in their previous one specifically because the ethics-related assignments were more interesting.

We saw that for many students this intervention increased engagement with assignments in addition to exposure to ethical concepts. We also already know from prior work that students are more likely to engage with computing if it is connected to the real world [13, 20] and that an emphasis on social impact of technology can help spark and sustain interest among underrepresented groups in particular [16]. Moreover, inclusion of ethics-related content around topics such as bias and diversity is a step towards increasing cultural competency among computing students, an important goal for both student retention and creation of a talent pool with a better understanding of equity and inclusion [33].

Finally, the majority of students in our survey sample are interested in seeing these types of assignments in their future programming classes. Our observations from this pilot intervention shows that there could, as suggested by a number of others [11, 23, 24, 30], be benefits for integrating ethics across the entire curriculum, and that replacing programming assignments is an effective strategy.

Ideally this paper has given others a sense for both the "why" and the "how" of our approach. One suggestion we have is to–as we did–start small. Replace one assignment to start, and the next semester, try another. Our experience with a complete overhaul in the Summer class was smoother due to having built up to that point, particularly because the INFO-B instructor was previously a TA for INFO-A. Finally, to provide guidance for two specific scenarios that might be challenging:

If you teach intro programming but are concerned that you don't know enough about ethics to do this on your own: Our assignments are available (see footnote 1), and others are creating them as well–such as Computer Science Professor Evan Peck's "Ethical Engine" CS1 programming assignment that incorporates the trolley problem (a classic moral philosophy hypothetical) into coding conditionals and another about bias in hiring algorithms [7, 21, 22]. So even if you do not have your own ethics experts to help, you can start with these kinds of open resources, which are becoming increasingly common [7, 23]. Note, however, that coming up with your own relevant context might not be as difficult as you think. There is a tech ethics scandal in the news nearly every week, and a number of available resources, including the syllabi for many standalone computing ethics classes that already exist [8]. You also might consider asking others for help–even beyond your own department.

If you know about ethics and want to help intro programming instructors: Replacing assignments with new ones is a great way to not be a huge burden on the instructors for the reasons we've already noted. Ask for copies of existing assignments and think about how the context could change. Even better, work directly with them if they can spare the bandwidth; instructors are often looking for ideas for new assignments.

Finally, based on the challenges we identified above, we will be making adjustments in the future ourselves, and have the following recommendations to help mitigate some of these problems:

(1) Be sure that when ethical concepts are integrated into assignments that it does not raise the difficulty of the assignments. Also consider starting with assignments that students do not traditionally struggle with; the less they struggle with the assignment, the more likely they are to be able to engage with the additional ideas.
(2) Particularly for technical topics that students struggle with, try introducing the ethical dilemma first through lecture or class discussion so that they have some time to grapple with it before having to worry about the code.
(3) Allow students to go over their solutions together so that they have an opportunity to discuss and reflect on their choices. This will both help them grapple with the ethical concepts and reinforce the technical material.
(4) Possibilities for evaluation include interview grading, which has the added benefit of providing personalized attention to students even at scale [12], and/or peer evaluation which has additional pedagogical benefits [5]. Short answers are also effective for a short reflection on design choices, and if there are problems with scale, pass/fail grading might be a solution.
(5) A more resource-intensive solution is to have one teaching assistant whose role is dedicated to helping with ethics content. This might include speaking in recitations and lecture, training other TAs, short answer grading, and/or helping to adjust the assignments. This option would be similar to Harvard's model of embedding Philosophy postdocs into CS classes [11].
(6) Additionally, prioritizing training for all TAs on the ethics content could result in learning benefits for the TAs as well as the students taking the class.

As we build out more assignments, our goal is to replace nearly all assignments in these large introductory programming classes with ethically contextualized content. Though we will take lessons from INFO-B (and also continue to teach it as fully ethics-integrated as a small summer class), we also recognize the constraints that come with large classes with auto-graded assignments and plan to design with those in mind, and to share broadly so that they can be used by others. We will also be conducting formal, survey-based evaluation that targets technical learning, ethical learning, and student engagement.

Finally, another advantage of introductory programming as a site for ethical context is that it can likely be easily adapted for K-12 computer science as well, which is appropriate because not only are children learning to code quite early, but they should also learn to be critical users of technology [6].

8 CONCLUSION

Nearly 25 years ago an NSF working group suggested integration of an ethics and social impact "tenth strand" throughout computer science core courses [19]. In recent years, large-scale projects like EmbeddedEthiCS [11] and the Responsible Computer Science Challenge [23], as well as individual models such as ethics integration into HCI classes [30] or machine learning [24], have shown more movement towards this goal. As universities grapple with how best to handle the increasing demand for computer scientists with expertise in ethically wrought fields such as artificial intelligence [10], our hope is that touching on ethics early and often, starting with introductory programming, can have a profound impact on the perception of ethics as part of "doing computer science."

9 ACKNOWLEDGEMENTS

This work was supported by Omidyar Network, Mozilla, Schmidt Futures and Craig Newmark Philanthropies as part of the Responsible Computer Science Challenge. Our thanks to Ioana Fleming and Murray Cox for allowing this intervention in their classes, as well as to all the teaching assistants who helped with implementation, with special thanks to Janet Ruppert and Tetsumichi Umada, and to Blakeley Payne for her valuable feedback.

REFERENCES

[1] Brett A. Becker and Thomas Fitzpatrick. 2019. What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students?. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. Minneapolis, MN, 1011–1017.

[2] Douglas Belkin. 2019. Colleges Mine Data on Their Applicants. https://www.wsj.com/articles/the-data-colleges-collect-on-applicants-11548507602

[3] Mary Elaine Califf and Mary Goodwin. 2005. Effective incorporation of ethics into courses that focus on programming. *ACM SIGCSE Bulletin* 37, 1 (2005), 347–351.

[4] Andy Cockbum and Neuille Churcher. 1997. Towards literate tools for novice programmers. *ACM International Conference Proceeding Series* Part F1293 (1997), 107–116.

[5] Maria de Marsico, Filippo Sciarrone, Andrea Sterbini, and Marco Temperini. 2017. Supporting mediated peer-evaluation to grade answers to open-ended questions. *Eurasia Journal of Mathematics, Science and Technology Education* 13, 4 (2017), 1085–1106.

[6] Daniella Dipaola, Blakeley H. Payne, and Cynthia Breazeal. 2020. Decoding design agendas: An ethical design activity for middle school students. *Proceedings of the ACM IDC Interaction Design and Children Conference* (2020), 1–10.

[7] Stacy A. Doore, Casey Fiesler, Michael S. Kirkpatrick, Evan Peck, and Mehran Sahami. 2020. Assignments that blend ethics and technology. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education, Extended Abstracts* (2020), 475–476.

[8] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What Do We Teach When We Teach Tech Ethics? A Syllabi Analysis. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education* (2020).

[9] Batya Friedman and Peter H. Kahn. 1994. Educating computer scientists: Linking the Social and the Technical. *Commun. ACM* 37, 1 (1994), 64–70.

[10] Natalie Garrett, Nathan Beard, and Casey Fiesler. 2020. More Than "If Time Allows": The Role of Ethics in AI Education. *Proceedings of the AAAI AIES Conference on AI, Ethics and Society* (2020), 272–278.

[11] Barbara J. Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: Integrating Ethics Broadly Across Computer Science Education. *Commun. ACM* 62, 8 (2019), 54–61.

[12] Dirk Grunwald, Elizabeth Boese, Rhonda Hoenigman, Andy Sayler, and Judith Stafford. 2015. Personalized attention @ scale. Talk isn't cheap, but it's effective. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education* (2015), 610–615.

[13] Mark Guzdial. 2009. Being Fluent with Information Technology. *Commun. ACM* 52, 5 (2009), 31–33.

[14] Rick Homkes, South Washington Street, and Robert A Strikwerda. 2009. Meeting the ABET Program Outcome for Issues and Responsibilities : An Evaluation of CS , IS , and IT Programs. *Proceedings of the ACM SIGITE Special Interest Group on Information Technology Education* (2009), 133–137.

[15] Scott Jaschik. [n.d.]. New Data on Admissions: Criteria That Matter, Early Decision and More. https://www.insidehighered.com/admissions/article/2018/11/12/new-data-admissions-including-application-trends-early-decision-and

[16] Nazish Zaman Khan and Andrew Luxton-Reilly. 2016. Is computing for social good the solution to closing the gender gap in computer science? *ACM International Conference Proceeding Series* 01-05-Febr (2016), 0–4.

[17] Stephanie Ludi, Matt Huenerfauth, Vicki Hanson, Nidhi Rajendra Palan, and Paula Garcia. 2018. Teaching Inclusive Thinking to Undergraduate Students in Computing Programs. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. Baltimore, MD, 717–722.

[18] C Dianne Martin. 1997. The Case for Integrating Ethical and Social Impact into the Computer Science Curriculum. *ITCSE Working Group Reports and Supplemental Proceedings* (1997), 114–120.

[19] C. Dianne Martin, Chuck Huff, Donald Gotterbarn, and Keith Miller. 1996. Implementing a Tenth Strand in the CS Curriculum. *Commun. ACM* 39, 12 (1996), 75–84.

[20] Hilarie Nickerson, Catharine Brand, and Alexander Repenning. 2015. Grounding computational thinking skill acquisition through contextualized instruction. *Proceedings of the ACM ICER Conference on International Computing Education Research* (2015), 207–216.

[21] Evan Peck. 2017. The Ethical Engine: Integrating Ethical Design into Intro Computer Science. https://medium.com/bucknell-hci/the-ethical-engine-integrating-ethical-design-into-intro-to-computer-science-4f9874e756af

[22] Evan Peck. 2018. Ethical Design in CS 1: Building Hiring Algorithms in 1 Hour. https://medium.com/bucknell-hci/ethical-design-in-cs-1-building-hiring-algorithms-in-1-hour-41d8c913859f

[23] Kathy Pham. 2020. Ethics and Social Responsibility in Computer Science Curricula. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education, Extended Abstracts* (2020).

[24] 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 (2019), 1–26.

[25] Carsten Schulte and Jens Bennedsen. 2006. What Do Teachers Teach in Introductory Programming?. In *Proceedings of the ACM Conference on International Computing Education Research (ICER)*. 17–28.

[26] Kimberley Scott and Xiaolong Zhang. 2014. Designing A Culturally Responsive Computing Curriculum For Girls. *International Journal of Gender, Science and Technology* 6, 2 (2014), 264–276.

[27] Ben Rydal Shapiro, Amanda Meng, Cody O'Donnell, Charlotte Lou, Edwin Zhao, Bianca Dankwa, and Andrew Hostetler. 2020. Re-Shape: A Method to Teach Data Ethics for Data Science Education. *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems* (2020), 124:1–13.

[28] Frank M Shipman and Catherine C Marshall. 2020. Ownership, Privacy, and Control in the Wake of Cambridge Analytica: The Relationship between Attitudes and Awareness. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. 533:1–12.

[29] Natasha Singer. 2018. Tech's ethical 'dark side': Harvard, Stanford, and others want to address it. https://www.nytimes.com/2018/02/12/business/computer-science-ethics-courses.html

[30] 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 ACM SIGCSE Technical Symposium on Computer Science Education*. Baltimore, MD, 940–945.

[31] Bernd Carsten Stahl, Job Timmermans, and Brent Daniel Mittelstadt. 2016. The Ethics of Computing: A Survey of the Computing-Oriented Literature. *Comput. Surveys* 48, 4 (2016), 1–38.

[32] O Tene and Jules Polonetsky. 2013. A Theory of Creepy: Technology, Privacy and Shifting Social Norms. *Yale Journal of Law & Technology* 16, 1 (2013), 1–32. arXiv:arXiv:1011.1669v3 http://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=2326830

[33] Alicia Nicki Washington. 2020. When twice as good isn't enough: The case for cultural competence in computing. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education* (2020), 213–219.