Repository Entry Template
Embedded EthiCS @ Harvard Teaching Lab

Overview	
Course:	CS 50: Introduction to Computer Science
Course Level:	Introductory
Course Description:	"Introduction to the intellectual enterprises of computer science and the art of programming. This course teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web programming. Languages include C, Python, and SQL plus HTML, CSS, and JavaScript. Problem sets inspired by the arts, humanities, social sciences, and sciences. Course culminates in a final project. Designed for concentrators and non-concentrators alike, with or without prior programming experience. Two thirds of CS50 students have never taken CS before. Among the overarching goals of this course are to inspire students to explore unfamiliar waters, without fear of failure, create an intensive, shared experience, accessible to all students, and build community among students."[1]
Module Topic:	Cultivating Moral Imagination with Value Sensitive Design
Module Author:	William Cochran
Semesters Taught:	Spring 2022
Tags:	Design [CS], stakeholders [both], direct and indirect stakeholders [phil], value [phil], moral imagination [phil], value sensitive design [phil]
Module Overview:	The main goal of this module was to prepare students to reflect on the ethics of their final projects. To that end, it introduced students to Value Sensitive Design as a framework for reflecting on the impact of their work for their projects' users. First, students were introduced to Value Sensitive Design through a short lecture, coupled with a toy example involving the deployment of new emoji. Then, students practiced applying this framework to two final projects from the previous semester's CS50 class. In small groups, students watched video presentations of these previous projects and then used whiteboards around the room to identify these projects' stakeholders (both direct and indirect), their values, and the impact these projects could have on stakeholders as they scale up. The module also included an overview of the various mini-modules that had taken place in prior weeks of the course.	For more information about these mini-modules, see the tabs below.
Connection to Course Material:	Students in CS50 complete a final project that tasks students with implementing the technical skills they have gained during the semester. This module (as well as several mini-modules which took place in prior classes) prepared students to engage with a section of their final project Design Document where they answered questions about the technical and ethical decisions they made when developing their projects.	This topic was chosen in consultation with the professor (David Malan) and preceptor (Carter Zenke). They had been looking for a way to integrate this ethical reflection into students' final projects for a while.

---

[1] https://cs50.harvard.edu/college/2022/spring/syllabus/

Goals

Module Goals:	1. Help students develop their capacity for moral imagination, especially when thinking about the broader impacts of their technical work. 2. Prepare students to reflect on the ethical decisions they will make in the development of their final projects. 3. Reinforce and solidify the general themes about morally responsible computing already touched on in previous ethics mini-modules (taught in Weeks 0, 2, 5, and 7).	One of the goals of CS50 is to empower students with new technical abilities that they can deploy in a wide variety of settings beyond their time in the course. The ethics lessons in this module—as well as in the mini-modules throughout this semester—sought to supplement these technical abilities with an ethics toolkit that could help them deploy their newly developed technical skills responsibly.
Key Philosophical Questions:	1. Do computer scientists have a responsibility to consider the moral and social impacts of their work? 2. At what point(s) do ethical values factor into the design process? 3. How can moral imagination help computer scientists think about the broader impacts of their work?	The mini-modules that took place in previous weeks sought to establish an affirmative answer to question 1, setting the stage for students to think about *how* to discharge this responsibility (questions 2 + 3).

Materials

Key Philosophical Concepts:	● Value sensitive design ● Stakeholders: direct and indirect ● Moral responsibility ● Moral imagination	Introducing students to the concept of both direct and indirect stakeholders helps students develop moral imagination by bringing into view the people that are potentially impacted by their technical design choices.
Assigned Readings:	● None	As with other classes in CS50, there was no assigned reading for this module; instead, the normal practice was to have students watch a video lecture recorded in the corresponding week of the previous Fall. In the week that this ethics module ran, students watched a lecture on emoji and representation (Week 10).

Implementation

Class Agenda:	1. Overview of the ethics mini-modules taught in prior weeks. 2. Introduction to Value Sensitive Design as a way to cultivate moral imagination and assess the	At the beginning of (2), I also used a comparison to developing a new building to help students visualize the way in which developing a new

	moral and social impacts of students' final projects. 3. Model applying the framework on an example involving the deployment of new emoji (from the Fall 2021 CS50 lecture, Week 10). 4. Small group activity*:* practice applying the framework on previous CS50 final projects. 5. Wrap up with class-wide debrief on small group work.	program needs to be sensitive to the (moral) landscape into which it is being built.
Sample Class Activity:	For the active learning portion of the module, students were put into small groups and tasked with applying the value sensitive design framework to two previous final projects from the most recent CS50 class (Fall 2021). After watching one of the videos, students wrote their responses on whiteboards around the classroom, identifying direct and indirect stakeholders, their values, and the likely impact on stakeholders and their values as the project scales up. Students then repeated the same process on another previous final project. Finally, there was a class-wide debrief where students discussed their in-group work.	Scaffolding the activity in this way—first seeing the instructor apply the framework with a toy example in class, then applying it in small groups—prepared students to eventually apply the same procedure on their own when working on their final projects. It should also be noted that the two past student projects used in this activity were chosen from among those posted on CS50's website. Prior to having their videos uploaded, students agree to having their videos posted and made publicly available.
Module Assignment:	The assignment connected with this module was integrated into the "Design Document" for students' final projects. In addition to responding to questions about the technical decisions students made (such as "What design challenge(s) did you run into while completing your project? How did you choose to address them and why?"), students were prompted to reflect on their ethical decisions as well. They were asked the following set of questions: ● What motivated you to complete this project? What features did you want to create and why? ● Who are the intended users of your project? What do they want, need, or value? (You should consider your project's users to be those who interact *directly* with your project, as well as those who might interact with it *indirectly*, through others' use of your project.) ● How does your project's impact on users change as the project scales up? You might choose one of the following questions to reflect on:	The hope is that by repeating this process multiple times with the appropriate scaffolding—in class, in small groups, on their own for their final projects—students will be able to conduct similar ethical analyses of the work they do even beyond their time in CS50.

- ○ How could one of your project's features be misused?
- ○ Are there any types of users who might have difficulty using your project?
- ○ If your project becomes widely adopted, are there social concerns you might anticipate?

Lessons Learned: In their module feedback surveys, students responded most favorably about the following: the interactive nature of the module (working in groups on the whiteboards), the concept of direct and indirect stakeholders, and thinking about the impact on stakeholders as the project scales up. Preparing the whiteboards for students to work on proved to be time-intensive, but students were highly engaged in the activity. It was most satisfying to see them apply what they learned during the activity to their own final projects.

About this module, one student wrote the following in their survey response: "I found it interesting that new software must be thought about in regards to the direct and indirect stakeholders--even something as simple as an emoji needs to be thought about in regards to how it affects others." Another reported that, "The ideas about scale were definitely interesting since some things only become real issues at larger scales."

Week 0 Mini-Module Repository Entry Template
Embedded EthiCS @ Harvard Teaching Lab

Overview

Course:	CS 50: Introduction to Computer Science
Mini-Module Topic:	Ethics as the Fourth Axis of Good Code
Mini-Module Author:	William Cochran
Semesters Taught:	Spring 2022, Week 0
Tags:	Design [CS], Recommender system [CS], value [phil]

Connection to Course Material:	At the beginning of CS50, students are taught three axes of good code: Correctness, Design, and Style. This mini-module adds a fourth axis, Ethics. Each axis comes with a question: (1) Correctness: does my code do what it's supposed to do? (2) Design: how well (qualitatively) is my code implemented? (3) Style: is my code readable/pretty*etc.*? (4) Ethics: what are the ethical implications of my code? When coding (or learning to code), often the question coders ask themselves is "can I do this?" The fourth axis prompts them to reflect on whether they *should*. After a brief introduction to Embedded EthiCS–what it is and why it's necessary—the mini-module prompts students to begin thinking about this large-scale question ("should I do this?") by asking smaller-scale questions about the ethics of a hypothetical scenario tied to the technical content from this first week.	Prompting students to think about the ethics of their work on the very first class session hopefully establishes a practice of thinking about the ethics of their work all through the semester. Over time, with more mini-modules in future class sessions, hopefully students will develop an intellectual habit of thinking about such questions even beyond their time in the course.

Goals	
Goals:	1. Introduce students to Embedded EthiCS—what it is and why it is necessary. 2. Prompt students to begin thinking about the ethical implications of their work.

Materials	
Assigned Readings:	● None

Implementation		
Agenda:	1.	Ethics as the fourth axis of good code.
	2.	Why Embed EthiCS?
	3.	Small group activity
	4.	Whole class debrief
Activity:		Hypothetical: "Say we're asked to implement a contacts application, as we've seen before in lecture. But we've also been asked to create a feature that recommends the users we call most frequently at certain times of the day: what are some questions we could ask about our code to determine if we should do this?"
		In small groups with the people at your table, take about 5 minutes to think of some questions that you could ask about your code to determine whether you should create such a feature.

Week 2 Mini-Module **Repository Entry Template**
Embedded EthiCS @ Harvard Teaching Lab

Overview	
Course:	CS 50: Introduction to Computer Science
Mini-Module Topic:	Trustworthy Computing
Mini-Module Author:	William Cochran
Semesters Taught:	Spring 2022, Week 2
Tags:	Compilers [CS], Hacking [CS] Trust [Phil], Trustworthiness [Phil]
Connection to Course Material:	By this point in the semester, students have learned some of the basics of coding, including: writing code in a programming language, importing libraries, and running compilers. This module prompts students to reflect on the social aspects of even this work. First, students are shown how an undetectable Trojan horse could be inserted into a compiler program and unwittingly added to their own code. How can they ensure that the code they rely on is reliable? They discuss possible responses, such as updating the criminal code and/ or creating social stigma around these practices. Finally, they undergo an activity designed to develop trustworthiness.

Goals	
Goals:	1. Prompt students to reflect on the social aspects of writing code. 2. Practice developing trustworthiness.

Materials		
Assigned Readings:	● None	The inspiration for this mini-module comes from Ken Thompson's "Reflections on Trusting Trust" ACM Turning Award Lecture (1984). It is short and relatively accessible, so it could work well as a pre-reading for students.

Implementation		
Agenda:	1. Crowdsource examples of places in their code when they rely on the outputs of other people (e.g., libraries, compilers). 2. Run through an example of how undetectable malicious code (e.g., a Trojan horse) could be unwittingly inserted into one's own code. 3. Discuss different possibilities for addressing this problem. 4. "Blind Snake" Trust-building exercise. 5. Debrief	
Activity:	"Blind Snake" Trust Exercise. Requires 5-6 volunteers. Must be willing to stand in a line, front to back, with others, arm's length apart. Have volunteers line up at a certain point in the room. The group's goal is to maneuver their way together to another point in the room (such as to another table, out of the classroom door, *etc.*) *The catch*: the group can only communicate by tapping the person ahead of them on the shoulder. - Tap the left shoulder to turn the person left, tap the right shoulder to turn the person right - Tap on both shoulders to go forward. - Everyone, except the person in the back, must keep your eyes closed the entire time.	This is a relatively common team-building activity, and many examples and different iterations can be found on the web. Students in the line must rely on each other to achieve their goal. Each must give reliable directions (outputs) to the person in front of them in line and rely on the instructions (inputs) they receive from the person behind them. The activity dramatizes the kind of trust and trustworthiness that goes into writing good code.

Set a timer for 60 seconds and see how far they get.

Week 5 Mini-Module **Repository Entry Template**
Embedded EthiCS @ Harvard Teaching Lab

Overview

Course:	CS 50: Introduction to Computer Science

Mini-Module Topic:	Factoring Ethics into Technical Trade-offs
Mini-Module Author:	William Cochran
Semesters Taught:	Spring 2022, Week 5
Tags:	data structures [CS], privacy [phil], accessibility [phil], decision-making [phil], trade-offs [both]

Connection to Course Material:	At this point in the course students learn about different data structures. This week's class presents students with a hypothetical scenario where they have to choose the best data structure for a specific task. First students work through the technical trade-offs involved. Then, for their ethics mini-module, students consider the ethical implications of their choice and deliberate about whether to change their decision, and why.

Goals

Goals:	1. Prime students to see that ethical considerations can factor into technical trade-offs. 2. Give students practice discussing how ethical considerations impact decisions about technical choices.

Materials

Assigned Readings:	● None

Implementation

Agenda:
1. Introduce hypothetical scenario [see Activity below]
2. Have students think through the technical trade-offs involved in using different data structures (linked lists vs. tries) for designing a specific app.
3. Short preamble on privacy and accessibility.
4. Small group discussion: (1) How do you think the use of linked lists vs. tries impacts user *privacy*? (2) How do you think the decision to use linked lists vs. tries impacts users' *access* to this piece of technology?
5. Debrief and discuss: did your ethical analysis change your original decision about which data structure to implement in this scenario? Why or why not?

Steps 1-2 are part of the regular class time. The mini-module begins at step 3.

Generally speaking, linked lists are faster to implement, but make the app run slower. Tries, by contrast, take longer to implement, but result in a faster-running app.

Activity:
Imagine you work for a company that has created a personal digital assistant that runs on a mobile device's OS. Customer reports lead you think that:

1. The assistant often doesn't recognize its wake word, especially when users have accents.
2. It is easily activated by accident, potentially putting clients in embarrassing situations.

The fastest way to solve the issue is to program the device to record short segments of users' speech and then hire reviewers to analyze what triggers the system's errors.

Your task: develop the best storage system to solve this problem.

Note - "best" in this case can be understood along several different axes.

Technical axes:
1. Time to run (how fast the app runs)
2. Time to develop (how long it takes to implement the solution)

Ethical axes:
1. Privacy
2. Accessibility

Generally speaking, linked lists are worse for privacy because they store the whole phrase. Tries, by contrast, segment user speech into separate syllables, making it harder to identify individual users' speech.

However, since tries take longer to develop, delaying the time to implement a solution means that people with accents will continue to experience difficulty using the app.

Overview	
Course:	CS 50: Introduction to Computer Science
Mini-Module Topic:	Ethical Design Principles
Mini-Module Author:	William Cochran
Semesters Taught:	Spring 2022, Week 7
Tags:	Databases [CS], Design [CS], Design principle [CS], Ethical design principles [phil], Facial Recognition Software [CS], Justice [Phil]
Connection to Course Material:	By this point in the course, students have seen how databases ideally conform to certain technical design principles, such as 'minimize redundancies' (to increase efficiency), and 'permit adaptability' (to make it easier to add new data). But reflecting on how databases are used 'in the wild' can lead us to think of other design principles that databases should adhere to as well. The case described in "Wrongfully Accused by an Algorithm" (NYT) exemplifies how design decisions about databases can have profound impacts on real people. Students read excerpts from the case and work together in small groups to draft an ethical design principle that would have prevented what happened in this case.

Goals	
Goals:	1. Raise awareness about the real world repercussions of poor database design. 2. Prompt students to practice thinking about ethical design principles.

Materials	
Assigned Readings:	● None

The case for this mini-module comes from "Wrongfully Accused By an Algorithm" (NYT), by Kashmir Hill. It would make for a good pre-reading assignment. Due to time constraints, students iln this mini-module, read excerpts.

Implementation

Agenda: 1.   Introduce students to the case of Mr. Williams described in "Wrongfully Accused by an Algorithm" (NYT)

          2.   Small group activity: students develop an ethical design principle based on their reading of the case.

          3.   Debrief and discuss with the whole class.

Activity: Now that you have read the excerpts from "Wrongfully Accused by an Algorithm," work with the students at your table to create a design principle that would have avoided the issues that poor database design created in this case.

Your principle should be of the following form:

When designing databases, data practitioners should strive to create databases that

_____, in order to

_____.

Once you've come up with your principle, write it on your table's Google slide.

Be prepared to share your group's work, too.