# The Poetry of Computer Science, the Computer Science of Poetry: Philosophy of Computation

CS 198: Fall 2017 DeCal Syllabus
Facilitators: Jongmin Jerome Baek, Alexandra Tran, and Dawnia Brody
Faculty Sponsor: Professor Anca Dragan

Philosophy of Computation at Berkeley
pocab.org

## 1    Course Motivation

> Computer science is no more about computers than astronomy is about telescopes.
>
> (Edsger Dijkstra)

When the word "computer science" is uttered, few people think of philosophy or poetry. Similarly, few people know that the "father of computer science", Alan Turing, was also a philosopher, or that the "mother of computer science", Ada Lovelace, was also a poet. Amongst the glamour and wads of cash at nearby Silicon Valley, the history of computer science is rarely discussed, and the motivations of major thinkers frequently forgotten. This is a shame, because what motivated these thinkers were in fact not money nor glamour but some of the most universal, basic, visceral questions such as

> I propose to consider the question, can machines think?                    (Alan Turing)

> [I will have] the most *harmoniously* disciplined troops; – consisting of vast *numbers*, and marching in irresistible power to the sound of *Music*. Is not this very mysterious?...But then, *what* are these *Numbers*? There is a riddle –                    (Ada Lovelace, at her deathbed)

The neglect of philosophical motivations is doubly a shame because, decades since its founders' deaths, computer science has steadily advanced its light, and we are now in a much better position to start answering these questions! Some ideas from computer science, such as uncomputability, P vs. NP, and quantum computing, have slowly seeped into such diverse areas of thought as child development, language, evolution, culture, epistemology, metaphysics, morality, and on and on. But the idea that computer science has philosophical roots and implications has been slow to enter the cultural zeitgeist.

We lament not so much the lack of discussion on the philosophy of computation, but the lack of *awareness* that such discussions are possible. Because of the lack of awareness, computer science is instead frequently viewed as materialistic, elitist, and a means to an end. We want to foster an academic culture where a sizable portion of students are actively talking about the philosophy of computation, so that computer science becomes more of an end in itself. We believe such a culture will not only provide a major motivation to study computer science for many more people, but also make the department more welcoming to people of diverse backgrounds, diverse viewpoints, and diverse areas of focus and competence.

The interdisciplinary nature of this course will stimulate and challenge both STEM and humanities students, synthesizing both kinds of knowledge. At the end of the semester, students will take with them much more than the content they've learned from the readings. Through thoughtful reflection and lively class discussions, each student will develop an analytical way of perceiving and thinking about the world, as well as their own personal philosophy. As we get to know like-minded students from diverse backgrounds, we will all gain a broader awareness of the state and the future of the field. The final project provides a unique opportunity to create original content in whatever medium each student is versed in, whether technical, lyrical, or philosophical.

# 2 Schedule

The class is organized into three clusters. Each cluster will take three to four weeks of discussion.

## 2.1 Cluster 1: The Computational Perspective

In this cluster, we provide a solid foundation about what it means to think of the world in terms of computation. We will read the excellent *The Information: A History, A Theory, A Flood* by James Gleick for setting of history and context. Interested readers may also refer to *Logicomix: An Epic Search for Truth* by Doxiadis & Papadimitriou for a dramatic portrayal of the ground-shaking discovery that a great number of things – indeed most things – are not provable. We will also read excerpts of *Why Philosophers Should Care About Computational Complexity* by Scott Aaronson.

- Week of September 11: Introductions, Class Policies, Philosophical thinking toolkit, Paradoxes

- Week of September 18: History and The Perspective: First part of *The Information*

- Week of September 25: History and The Perspective: Second part of *The Information*

- Week of October 2: *The Information* recap, Computation and Information

- Week of October 9: Uncomputability and Intractability: Parts of *Logicomix* and *Why Philosophers Should Care About Computational Complexity*

## 2.2 Cluster 2: Society, Memes, Culture, Language...

In this cluster, we apply the computational perspective to topics not often explored with computation. We will talk about an information-theoretic view of society, memes, philosophy of language, neural embeddings of words (word vectors) and their philosophical implications, how to write programs that write poems, and a possible interpretation of culture in terms of computation. Readings include excerpts from *Incomplete Nature: How Mind Emerged From Matter* by Terrence Deacon, *From Molecule to Metaphor* by Jerome Feldman, some research papers in the biological computation, *Culture, Dialectics, and Reasoning about Contradiction* by Kaiping Peng, and *Multimodal Distributional Semantics* by Elia Bruni.

- Week of October 16: Society and Memes: Weber, Durkheim, Exponents; Formal theory of memes, understanding its informal manifestations

- Week of October 23: Culture: *Culture, Dialectics, and Reasoning about Contradiction*, *The Geography of Thought*

- Week of October 30: Poetry/Natural Language Processing: *Multimodal Distributional Semantics*

## 2.3 Cluster 3: Free Will, AI, AI Ethics

Our last cluster addresses that question posed by Lovelace and Turing all those years ago: *Can machines think?* We will talk about both sides of the debate, whether the "AI scare" is just overblown hype or not, whether we can and how we would program a moral AI. Readings include parts of *A Turing Test for Free Will* by Seth Lloyd, *Universal Knowledge-Seeking Agents* by Laurent Orseau, *Moral Philosophy Will Be Part of the Tech Industry* by Stuart Russell, and *Gödel, Escher, Bach* and *The Mind's I* by Douglas Hofstadter. The cluster ends with a discussion of what free will might have to do with uncomputability.

- Week of November 6: Strong vs. Weak AI: parts of *Minds, Brains, and Programs*, parts of *Gödel, Escher, Bach* and *The Mind's I*

- Week of November 13: Free Will and AI ethics: *Universal Knowledge-Seeking Agents*, *Moral Philosophy Will Be Part of the Tech Industry*, *A Turing Test for Free Will*

## 2.4 Project Presentations

The last two weeks of class will be dedicated to presenting your projects!

# 3 Grading and Logistics

All majors and grade levels are welcome. This course is worth 2 units, and we will meet once a week for 80 minutes. Attendance, active participation, and completion of reading assignments is required.

This is a reading-intensive course; over the course of the semester, you will read no less than 500 pages, mostly fairly accessible journalism-style writing and a few research papers. You may have to purchase no more than two books. Each week, you will write a short essay (200-500 words) responding to a topic discussed in class and in the readings.

The class will culminate in a final project presentation, for which you can either (1) write a collection of poems about computer science, or (2) a program that writes poetry, or (3) a long (1000-2000 word) essay or philosophical dialogue about an open research question, such as, "If a self-driving car has to kill this person or that person, who should it kill, and how should it make the decision?" "Are humans universal Turing machines?", etc.

## 3.1 Detailed Grade Breakdown

- Attendance and participation: 33%

- Weekly assignments: 33%. Each assignment is worth 4% of your grade, and there will be 9 assignments (thus you could, in principle, gain 36% of your grade through assignments).

- Final project: 33%

To pass the course, students must fulfill at least 71% of the above.

## 3.2 Grading Details

We do not expect you to have taken a philosophy class before (though it would be great if you have!). You will get a perfect score on your assignments as long as you show a genuine engagement with the topic—thoughtfulness in your words, openness toward others, and a willingness to fully participate. We are always available through email or office hours to answer questions, discuss difficult concepts, and address any other concerns.

Of the three project options, (1), a collection of poems about computer science, and (2), a program that writes poetry, will be graded on a much higher bar than (3), a long (1000-2000 word) essay or dialogue about an open research question. For (1), a certain level of poetic mastery will be required; for (2), a certain level of technical mastery will be required. Choose (3) unless you have a specific, long-held vision and expertise to tackle (1) or (2).

## 3.3 Course Policies

Late work can be turned in for half credit up to 48 hours after the assignment's due date. Please contact a facilitator via email or in person to excuse an absence. Phones and laptops are not allowed in class. Also, please don't plagiarize work.

Refer to Campus Policies and Guidelines regarding religious, extracurricular, and emergency conflicts with academics. Please communicate with the facilitators regarding DSP accommodations. Disclosure is not necessary, but please let us know the best way to fulfill your needs.

## Contact Information

Please feel free to email us at pocabdecal@gmail.com
Office Hours: Thursdays 2:15-4PM at Cafe Strada, and by appointment
Philosophy of Computation at Berkeley is a student organization dedicated towards computer science as an end in itself, software engineers who care deeply about ethics, and radically interdisciplinary thought.