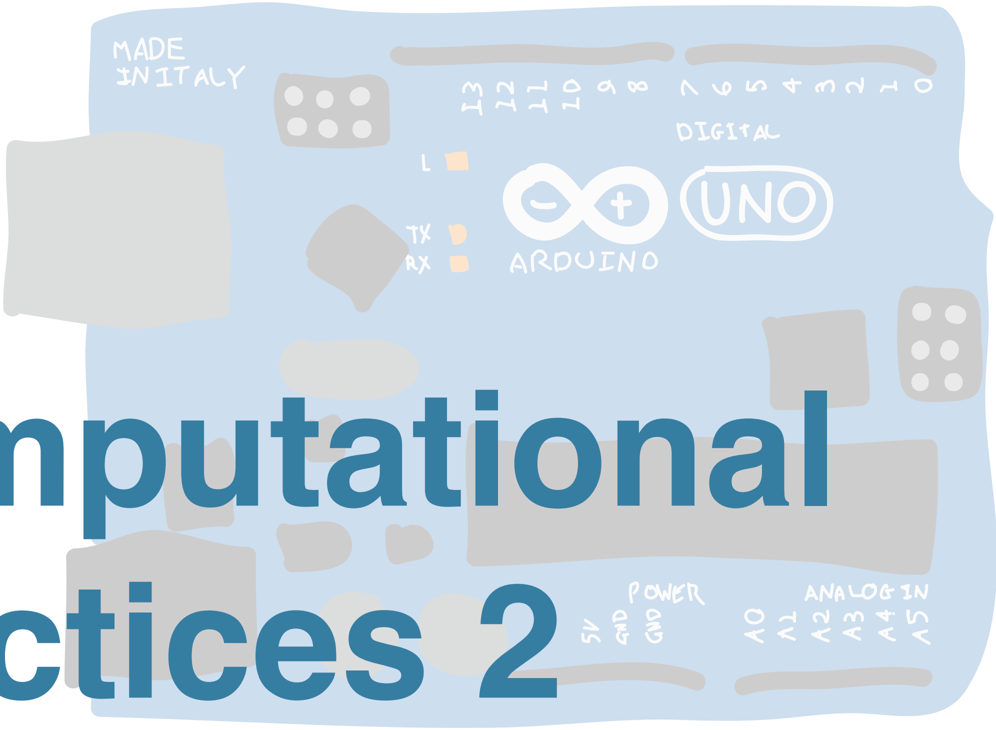

Computational Practices 2

California College of the Arts
SCIMA-300 Computational Practices 2
January 22 – May 6, 2019
Wednesday, 7:15-10pm
SF Main Campus, Room 107 (Hybrid Lab)

Instructors: J.D. Zamfirescu-Pereira jzamfirescupereira@cca.edu
Rolf Widenfelt rolf.widenfelt@cca.edu
Office hours: by appointment

Goals

Building on the skills and approaches developed in Computational Practices 1, this course will extend your depth of knowledge in computation through a series of case studies, deconstructions, and mini-projects. Movement, vision, reaction, sensing, patterning, repetition, and emulation are processes we can encode using computer code, electronic components, and mechanical parts, enabling the creation of robots, sculptures, visualizations, conversations, simulations, and experiments capable of illuminating the world and discovering deeper truths.

We will deconstruct work by contemporary artists, designers, and researchers to better understand the design and implementation of software and hardware systems. Topics typically include event-driven programming, APIs and data handling, web servers, motor control, Serial communication, wireless communication, printed circuit board design, and advanced debugging. Through prototypes, you will exercise these new tools and techniques. In a final project, you will define, research, and see to completion a technically advanced piece to be exhibited at a year-end show, suitable for inclusion in a portfolio.

Expect to have fun, and expect to be challenged. Expect to think in ways you might not be familiar or comfortable with, and expect to find yourself more capable than you thought.

Topics

Topics will span advanced concepts in programming and electronics; individual topics covered will depend on student interest.

Class Repository

http://github.com/zamfi/cca-computational-practices-2-spring-2020

The class repository will be used to assign homework and reading assignments, and will contain useful code, libraries, and links to other resources. Github allows you to receive email updates when a repository changes — take advantage of this feature!

Required Textbook

No textbook is required, but there are many good online references and a few good books out there we can recommend — it might be useful to get a book as a reference.

Prerequisites

- 200-level (or equivalent) experience with programming & electronics. If you are unsure, you should find this diagnostic test not especially challenging: https://git.io/vinuK
- Algebra. You should be comfortable with simple equations, manipulating numbers, fractions, solving for variables, and the engineering prefixes (micro, milli, kilo, mega, etc.)
- Some understanding of geometry & trigonometry will also be helpful.

Requirements and Expectations

- **Always bring your computer to class**. If you don't have a computer, check one out from the Media Center. You will need a computer to participate, and your grade will be negatively affected if you don't bring one.

- **Participate in class!** Ask questions, guess answers, propose topics, share interesting projects you've found, push the envelope, explore your interests, and teach us! There are no stupid questions; admitting when you don't know something should be a point of pride. Chances are you're not the only one with a question, just the bravest one. *As outlined in the CCA Student Handbook, attendance in class is mandatory and three or more unexcused absences will result in a failing grade.*

- **Plan to spend 3-9 hours a week on homework.** If you don't have a solid foundation in math you may need more time. Plan ahead!

- **Submit your homework on time.** Show your work in homework and exams to receive full credit. Write clearly and legibly. Attend class, and don't be late. These are not requests, these are requirements.
- **You are responsible for checking your email** and the class repository for updates.

Access & Wellness Services

CCA says:

> Students with disabilities, including disabilities that are not clearly evident like chronic diseases or learning disabilities are encouraged to notify their instructor after class or during office hours. CCA will make reasonable accommodations for persons with documented disabilities. Students should contact Suzanne Raffeld, Director of Access and Wellness Services (email: sraffeld@cca.edu; phone: 510.594.3775), to answer any questions or for assistance. For more information, consult CCA's webpage at: http://www.cca.edu/students/resources/disability.

We want you to succeed in this class. Please make use of the Hybrid Lab coaches, the Learning Resource Center, and talk to me if you feel you are struggling with the material — we can help you do better in this class, but only if you come talk to us!

Tentative Course Outline

Weeks 1-3: Machine Learning and Artificial Intelligence
Weeks 4-6: React and Apps on the Web
~~Weeks 7-9: Electronics, Robots, and Remote Control~~
Weeks 10-14: Final Project

In addition, throughout the term, we will discuss and assign homework on specific advanced topics.

Grading

We will consider the various components of the class in roughly the following proportions:
50% Homework & Assignments
40% In-Class Lab & Project Work
10% Attendance & Participation

Grading Rubric

A: Excellent. You've exhibited exemplary conceptual, technical and perceptual ability implementing projects. You've demonstrated a deft understanding of required readings, and ability to successfully communicate ideas and processes to others. All work is commented and clearly demonstrates understanding of each lesson.
B: Good. You've completed assignments, and demonstrated a grasp of most of the main aspects of each lesson, but not all. You're able to communicate information, and step by step processes well. In projects, conceptual, perceptual, and technical skills are present.
C: Satisfactory. You've completed the assignments but may lack enthusiasm or drive to push the work into detailed display of comprehension. You have not demonstrated comprehensive knowledge of the application or programming environment.
D: Unsatisfactory. You have not completed the work as assigned. Substantial problems exist in your work.
F: Fail. You did not submit work, or work is below unsatisfactory level.

Words of Advice

Exploration is a key part of this class. You'll get more out of your work if you give yourself extra time and have patience. Sometimes you'll hit a dead end and have to start over — don't despair, it happens to everyone, but give yourself extra time just in case.

This class should be both fun and intense. It's most fun when you enjoy what you're working on, so make a point of taking some time to explore projects you make want to work on in the future!

CCA Learning Outcomes

Students who successfully complete their SCIMA200 requirement will be able to:

1. Demonstrate observational skills — in CP2, through effective debugging by observation of program behavior.
2. Learn through systematic experimentation — in CP2, through effective debugging by experimenting with solutions to possible problems.
3. Recognize the difference between inductive and deductive reasoning, and apply them appropriately to understand a given or observed set of facts — in CP2, through different logical program designs and structures.
4. Use modeling strategies to represent the possible outcomes of a series of observations or experiments or to interpret complex data — in CP2, through effective debugging by modeling program execution.
5. Draw explicit connections between SCIMA courses and their creative work in their majors — in CP2, through your final project.
6. Articulate the importance of the learning they've done in their SCIMA coursework to their understanding of at least one sociocultural

phenomenon beyond CCA — in CP2, through interactions with the broader technology community.

SCIMA300 level courses address the same LOs but are distinguished by the increased complexity of the problems posed and phenomena studied. 300 level students will be asked to resolve those problems or explain those phenomena by integrating multiple facts, patterns, or data sets, and/or by generating independent strategies for approaching unfamiliar problems or situations.

Courses in the Computational Practices minor have additional learning outcomes; this course addresses the following two:

1. Students should have a conceptual and practical knowledge of programming for screen-based and physical computing, and the ability to apply that knowledge to individual art/design practice.
2. Students should be able to communicate and collaborate on technology-reliant projects with other artists and designers, as well as potential peers or professional collaborators with technical backgrounds.