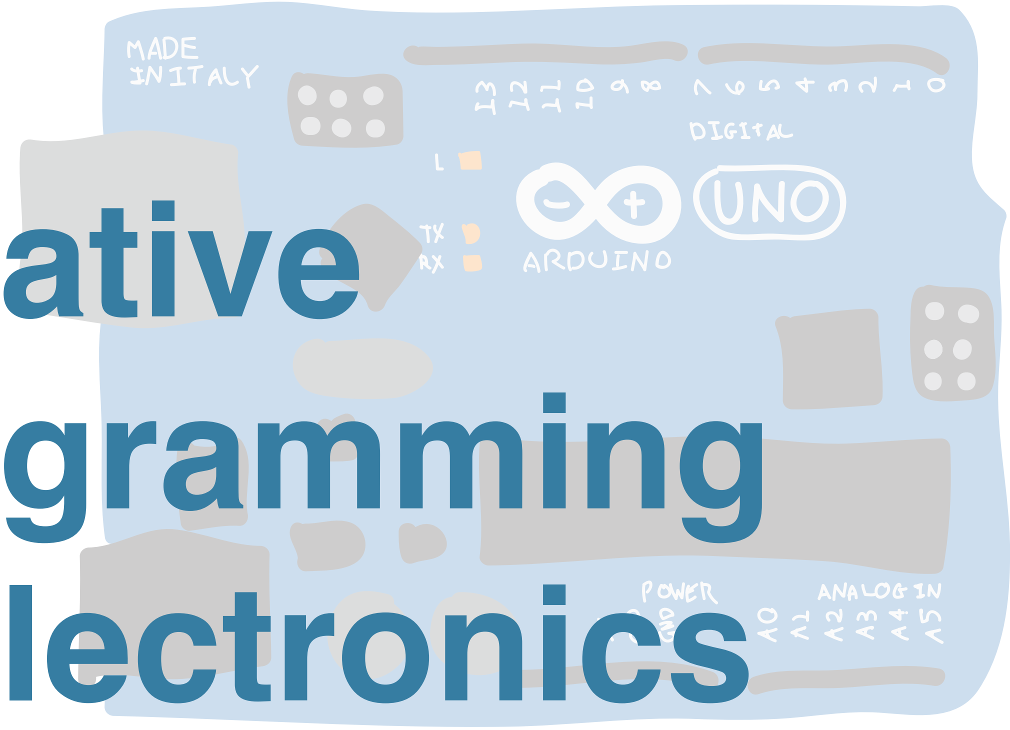
Creative Programming & Electronics



Jacobs Institute @ UC Berkeley
DES INV 23: Creative Programming & Electronics
January 18 – May 3, 2017 (no class March 29)
Thursdays, 10am – 1pm
Jacobs Hall, Room 210

Instructor:	J.D. Zamfirescu-Pereira	zamfi@berkeley.edu
GSI:	Noura Howell	noura@berkeley.edu

Office hours: Thursday, 1pm-2pm, 3rd floor Jacobs (J.D.)
Monday, 11am-12pm, Virtual (J.D.)
Monday, 1:30-2:30pm, 3rd floor Jacobs (Noura)

Course Web Page

bCourses: https://bcourses.berkeley.edu/courses/1470326

GitHub: https://github.com/zamfi/ucb-jacobs-creative-programming-electronics-spring-2018

All course assignments—homework and readings—will be listed on both course webpages, along with contain useful code, libraries, and links to other resources. GitHub and bCourses both allow you to receive email updates when a repository changes — take advantage of this feature!

Catalog Description

This course teaches techniques to conceptualize, design and prototype interactive objects. Students will learn core interaction design principles and learn how to program devices with and without screens, basic circuit design and construction for sensing and actuation, and debugging. Students work individually on fundamental concepts and skills, then form teams to work on an open-ended design project that requires a synthesis of the different techniques covered.

Prerequisites

This course has no pre-requisites. This course pairs well with DES INV 22: Prototyping & Fabrication I, but the two courses can be taken in either sequence, or individually.

Enrollment Policy

This course is designed for lower division undergraduate students from any department. Upper division students may be admitted if room allows.

Required Materials

- A custom kit of electronics, designed for this course ($100)
- Maker Pass for accessing prototyping and electronics labs in Jacobs Hall and the CITRIS Invention Lab ($75)
- A personal laptop for the programming and electronics assignments and in-class work (Windows, Mac, or Linux all ok). **Always bring your laptop to class!**

Course Goals & Objectives

This course explores programming and electronics by solving puzzles, building sketches in p5.js using JavaScript, and prototyping robots using the Arduino microcontroller platform.

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

By the end of the course, you will:

- Understand the design space of interactive objects, learn design principles for structuring user interaction with such devices, and be familiar with common design patterns and anti-patterns.
- Be able to write and debug programs for interactive, event-driven input and output on screens as well as programs that implement an interactive input-computation-output loop in hardware.
- Understand how to construct and measure basic electronic circuits.
- Understand how to read data sheets, specify and select electronic components, and integrate several components into a working system.
- Understand operating principles and design tradeoffs behind different types of sensors and actuators.
- Know how to connect sensors and actuators to microcontrollers, and communicate with them from embedded code.
- Use wired or wireless communication protocols to exchange information between interactive devices and other computers.
- Construct systems that offer user interactions which span both graphical user interfaces and embedded computation.
- Know how to construct project enclosures using digital fabrication tools.
- Leverage this technical toolbox to achieve novel interactive experiences.
- Know how to teach yourself to use more advanced tools and techniques after you've completed this course.

Readings

No textbook is required. The following readings can be accessed either through UC Berkeley's online eBook subscription, or through reserves in the engineering library:

- Lauren McCarthy, Getting Started with p5.js
- Daniel Shiffman, Learning Processing, 2nd edition
- Paul Scherz, Practical Electronics for Inventors, 4th edition
- Charles Platt, Encyclopedia of Electronic components, Vols 1 and 2
- Simon Monk, Programming Arduino, 2nd Edition
- Tom Igoe, Making Things Talk, 3rd Edition
- Forrest Mims, Getting Started in Electronics

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.
- **Plan to spend 3-9 hours a week on homework.** If you don't have a solid foundation in high school math you may need more time. Plan ahead!
- **Submit your homework on time.** Show your work in homework and assignments 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.

Grading

We will consider the various components of the class in roughly the following proportions:

20% Attendance & participation / in-class labs & workshops
40% Individual homework & assignments
10% Team evaluations
30% Final project

Tentative Course Outline

Week	Topic	Activity or Assignment
1	Course Overview & Introduction. Basics of programming: functions, variables, conditions, and loops.	Sign up for Maker Pass, complete basic workshop safety training. Writing programs without code.
2	Writing code: structure and syntax of JavaScript.	Write code to solve puzzles and draw shapes; read about p5.js.
3	Graphical output: designing and drawing to a screen.	Writing & stepping through code

Week	Topic	Activity or Assignment
4	Programming for interaction: getting user input & mastering the input -> computation -> output loop. Visual & audio output.	Design Challenge 1: Make a visual musical instrument
5	Modeling data & simulation: arrays, objects, discrete time steps, events.	Design challenge, continued
6	Play test, critique of design challenge 1. Overview of Electronics: Arduino, voltage/current, Ohm's law, common components, schematics, breadboards.	*Design Challenge 1 due.* Building basic circuits.
7	Sensors & user inputs: analog vs. digital, associated circuits (pull-downs, voltage dividers). Touch sensors. Basic outputs: LEDs, LCDs. Importance of feedback for interactivity	Exercise: design a vocabulary of status and notification messages with only one pixel.
8	Motors & Motion: DC, servos, steppers. Vibration motors. Linkages.	Design Challenge 2: Robotic Lamp
9	Digital components: SPI, I2C, Serial. Data sheets & part selection. Libraries & APIs.	Design challenge, continued
10	Demonstration & critique of second design challenge. Interaction patterns and anti-patterns for interactive devices	*Design Challenge 2 due.* Final Project Design Brief: TBD
11	Systematic debugging, structuring programs and circuits for ease of development and maintenance.	*Final Project proposal due*
12	Robustness: from breadboard to protoboard to custom PCB; soldering.	
13	Critique of Alpha; open work time.	*Alpha version due*
14	Exploration: advanced topics; computer vision; audio synthesis.	*Insights from testing due* Plan for final iteration
15	Reading/recitation	Present during Jacobs Design Showcase
Final	Final Reports due	Online project documentation, including demonstration video

Words of Advice

Exploration is a key part of this class. Remember that learning how to learn is one of the most useful goals of this course; when you’re stuck, think about what approaches might help you make progress. Don’t hesitate to ask for help.

You’ll get more out of your work if you give yourself extra time and have compassion for yourself. This course should be both fun and intense. It’s most fun when you enjoy what you’re working on, so choose a final project that motivates you!