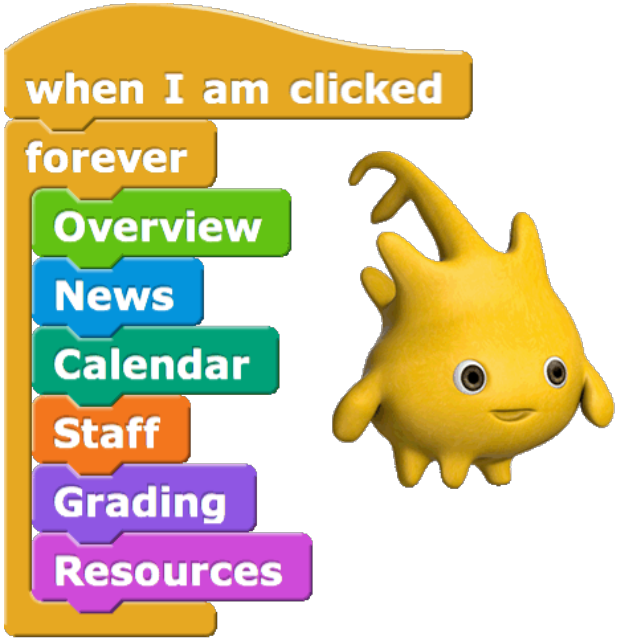


CS10
The Beauty & Joy of Computing
Summer 2015

Quick Links

SNAP!
Piazza
Labs
bCourses
Past Webcasts
Course Policies

Lab Check-Off Questions
Check Your Slip Days
Practice Exams

Updates

Please check this section for important updates!

- λ May 28, 2015: Welcome to CS10!
- λ May 28, 2015: This site is currently under development!

Overview

Our labs are held in the Apple Orchard, which is not only the newest lab on campus with the fastest machines, but also has the most natural light!

Fall 2009 students pair programming in Scratch.

CS10: *The Beauty and Joy of Computing*, is an exciting new course offered by the UC Berkeley EECS Department. Computing has changed the world in profound ways. It has opened up wonderful new ways for people to connect, design, research, play, create, and express themselves. However, just using a computer is only a small part of the picture. The real transformative and empowering experience comes when one learns how to program the computer, to translate ideas into code. This course will teach students how to do exactly that, using *Snap!* (based on Scratch), one of the friendliest programming languages ever invented. It's purely graphical, which means programming involves simply dragging blocks around, and building bigger blocks out of smaller blocks.

But this course is far more than just learning to program. We'll focus on some of the "Big Ideas" of computing, such as abstraction, design, recursion, concurrency, simulations, and the limits of computation. We'll show some beautiful applications of computing that have changed the world, talk about the history of computing, and where it will go in the future. Throughout the course, relevance will be emphasized: relevance to the student and to society. As an example, the final project will be completely of the students' choosing, on a topic most interesting to them. The overarching theme is to expose students to the beauty and joy of computing. This course is designed for computing non-majors, although interested majors are certainly welcome to take the class as well! We are especially excited about bringing computing (through this course) to traditionally under-represented groups in computing, i.e., women and ethnic minorities.

Some context: in the Fall of 2009, we piloted a 2-unit version of this course as the freshman/sophomore seminar CS39N: The Beauty and Joy of Computing to 20 students. It was such a success that we decided to move ahead to make this course our new computing course for non-majors, replacing the venerable CS3L and CS3S . Since then, this has been one of the most popular courses in EECS. Don't believe us? See for yourself! We're continuing to grow the course as word spreads to more students. We're continually replacing the weakest parts of the curriculum and hope you'll enjoy!

We will be using Pair Programming, described best by Laurie Williams, a computer science professor at North Carolina State University: "Two programmers working side-by-side, collaborating on the same design, algorithm, code or test. One programmer, the driver, has control of the keyboard/mouse and actively implements the program. The other programmer, the observer, continuously observes the work of the driver to identify tactical (syntactic, spelling, etc.) defects and also thinks strategically about the direction of the work. On demand, the two programmers can brainstorm any challenging problem. Because the two programmers periodically switch roles, they work together as equals to develop software."

Assignment Calendar

Semester Schedule (Subject to Change):

This is the first few weeks of summer schedule!

TENTATIVE exam dates are:
Quest: July 8 (in class, 1 hour)
Midterm: July 29 (evening / after class, 2 hours)
(This may also be July 22)
Final: August 12 (evening, 3 hours)

Reading Assignments Key:

- **Blue** readings are required,
- **Green** readings are required-but-challenging (understand the "big idea" concepts, rather than the technical details),
- ***Red*** readings are optional but recommended. Note that these readings are *NOT* tested.

Reading quizzes will be at the beginning of both the Monday and Wednesday labs based on the readings at the beginning of each row of the calendar.

Week	Dates	Readings (Sa/Su)	Lab 1 (M)	Discussion 1 (M/Tu)	Lecture 1 (M)	Lecture 2 (Tu)	Work Session (Tu/Th)
1	6-22 to	No	1: Welcome	Welcome to	Welcome and	Functions	Work

	6-26	Readings	to SNAP!	CS10!	and Abstraction	Functions	Session

Week	Dates	Readings (M/Tu)	Lab 2 (W)	Discussion 2 (W/Th)	Lecture 3 (W)	Lecture 4 (Th)	Assignments
1	6-22 to 6-26	λ Prof. Harvey's Intro to Abstraction λ TEDx Berkeley: The Beauty and Joy of Computing λ Why Software is Eating the World λ All I Really Need to Know about Pair Programming I Learned In Kindergarten λ *Learning to Code!* λ *Is*	2: Build Your Own Blocks **Reading Quiz 1 (in lab)**	Anatomy of a Computer and the Power of Binary	Creativity and Abstraction	Creativity	HW0 Submit *due Wed 6/24 at 11:59pm*

		Abstraction the Key to Computing? λ *Scratch: Programming for All*					
Week	**Dates**	**Readings (Sa/Su)**	**Lab 1 (M)**	**Discussion 1 (M/Tu)**	**Lecture 1 (M)**	**Lecture 2 (Tu)**	**Work Session (Tu/Th)**
2	6-29 to 7-3	λ BtB Chapter 1 λ Designing Games with a Purpose (GWAP) λ The Story of Alan Turing & His Machine λ *More readings on video games* λ *Animating a Blockbuster*	3: Functions **Reading Quiz 2 (in lab)**	Functions	Programming Paradigms	Algorithms	4: Lab: Finch Robots

Week	Dates	Readings (M/Tu)	Lab 2 (W)	Discussion 2 (W/Th)	Lecture 3 (W)	Lecture 4 (Th)	Assignments
2	6-29 to 7-3	λ BtB Chapter 2 λ What's Hot in the Art World? Algorithms λ How Algorithms Shape Our World λ Program or Be Programmed λ *Program or Be Programmed: A Guide*	5: Lists I **Reading Quiz 3 (in lab)**	All About Lists	Algorithmic Complexity	No Lecture	HW1: Word Guessing Spec | Submit *due Wed 7/1 at 11:59pm*

Week	Dates	Readings (Sa/Su)	Lab 1 (M)	Discussion 1 (M/Tu)	Lecture 1 (M)	Lecture 2 (Tu)	Work Session (Tu/Th)
		λ BtB Chapter 4					

3	7-6 to 7-10	Reading Segment 1 λ BtB Chapter 4 Reading Segment 2 λ *Living in a Digital World* λ *BtB Chapter 3*	6: Algorithms & Algorithmic Complexity **Reading Quiz 4 (in lab)**	Algorithmic Complexity and Quest Review	Recursion I	Social Implications I	Work Session

Week	Dates	Readings (M/Tu)	Lab 2 (W)	Discussion 2 (W/Th)	Lecture 3 (W)	Lecture 4 (Th)	Assignments
3	7-6 to 7-10	No Readings **Quest Review Session** Sunday, 7/5 2-4pm 306 Soda	7: Trees and Fractals using Recursion **Reading Quiz 5 (in lab)**		**Quest In Class**	Recursion II	HW2: Snowman Spec | Submit *due Mon 7/6 at 11:59pm* Midterm Project **Proposals** Spec | Submit *due Mon 7/6 at 11:59pm*

Week	Dates	Readings (Sa/Su)	Lab 1 (M)	Discussion 1 (M/Tu)	Lecture 1 (M)	Lecture 2 (Tu)	Work Session (Tu/Th)
4	7-13 to 7-17	λ BtB Chapter 5 Reading Segment 1 λ BtB Chapter 5 Reading Segment 2 λ BtB Chapter 5 Reading Segment 3 λ BtB Chapter 6 (27-37)	8: Recursive Reporters I **Reading Quiz 6 (in lab)**	Quest Debrief & HW Questions	Social Implications II	Concurrency	Work Session

Week	Dates	Readings (M/Tu)	Lab 2 (W)	Discussion 2 (W/Th)	Lecture 3 (W)	Lecture 4 (Th)	Assignments
		λ BtB Appendix λ TED: What is the					

4	7-13 to 7-17	Internet? λ Moore's Law Keeps Going, Defying Expectations λ Free Lunch is Over λ *How Moore's Law Works* λ *What is IBM's Watson?* λ *Brian Harvey's AI notes* λ *The First Church of Robotics* λ *Spending Moore's dividend (CACM)*	9: Concurrency **Reading Quiz 7 (in lab)**	Concurrency	Internet I	Internet II	HW3 Spec | Submit *due Mon 7/13 at 11:59pm*
		Readings		**Discussion**	**Lecture 1**	**Lecture 2**	**Work Session**

Week	Dates	(Sa/Su)	Lab 1 (M)	1 (M/Tu)	(M)	(Tu)	(Tu/Th)
5	7-20 to 7-24	λ BtB Chapter 7 λ The Heartbleed Bug λ When Servers Bleed	10: Practice with HOFs and Functions as Data **Reading Quiz 8 (in lab)**	The Internet	Lambda & HOFs	Data	Work Session

Week	Dates	Readings (M/Tu)	Lab 2 (W)	Discussion 2 (W/Th)	Lecture 3 (W)	Lecture 4 (Th)	Assignments
5	7-20 to 7-24	λ BtB Chapter 8	11: Tic Tac Toe **Reading Quiz 9 (in lab)**	Data & HOFs	Social Implications III	Besides Blocks I	Midterm Project Spec | Submit *due Mon 7/20 at 11:59pm* "Explore" Writing Assignment Spec | Submit *due Fri 7/24*

							at 11:59pm

Weekly Schedule

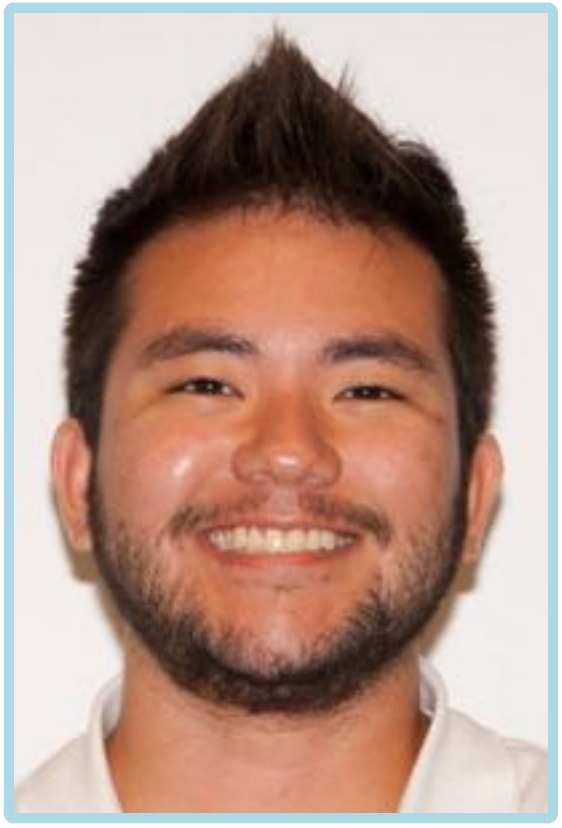
This calendar displays the class schedule for the *current* week. Click on any event to see the building location on a map.

Staff

Instructors

Instructor Michael Ball

michael@cs10.org



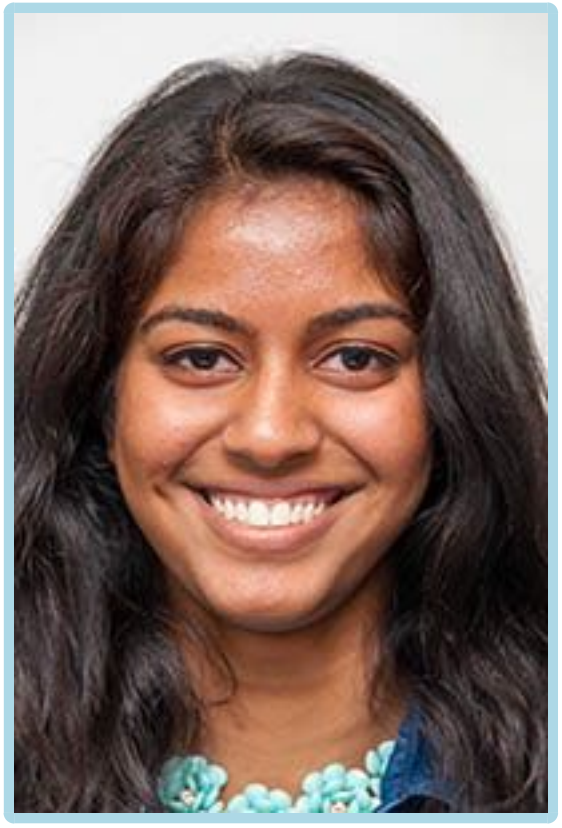
Instructor Jon McKinsey

jon@cs10.org

Teaching Assistants

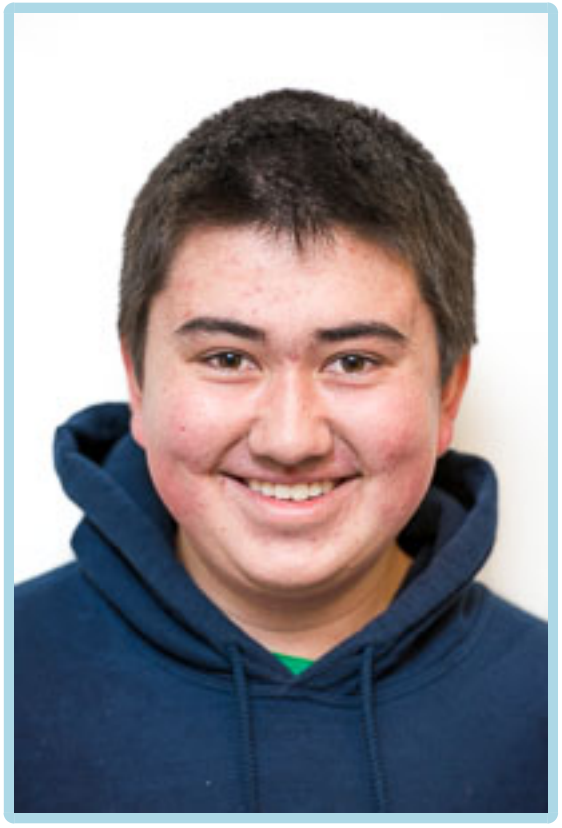
Head TA Lauren Mock

`lauren@cs10.org`



TA Arany Uthayakumar

arany@cs10.org

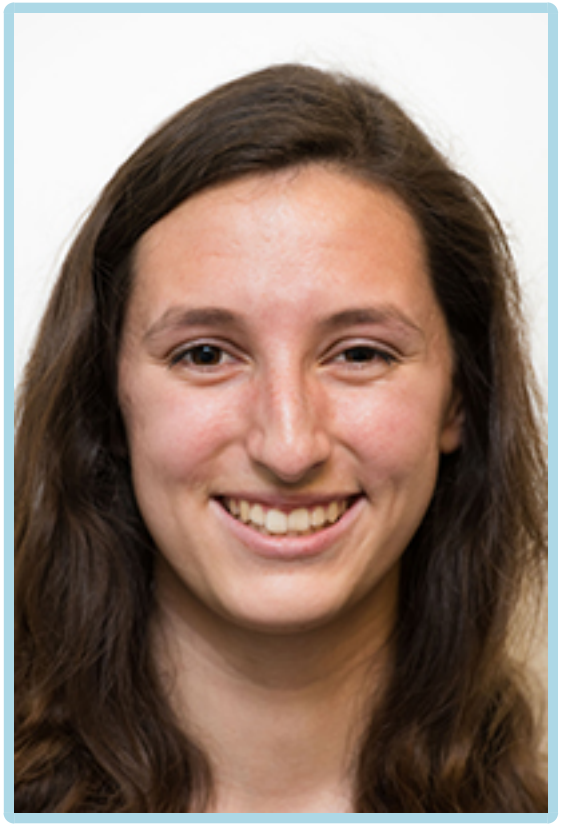


TA Erik Dahlquist

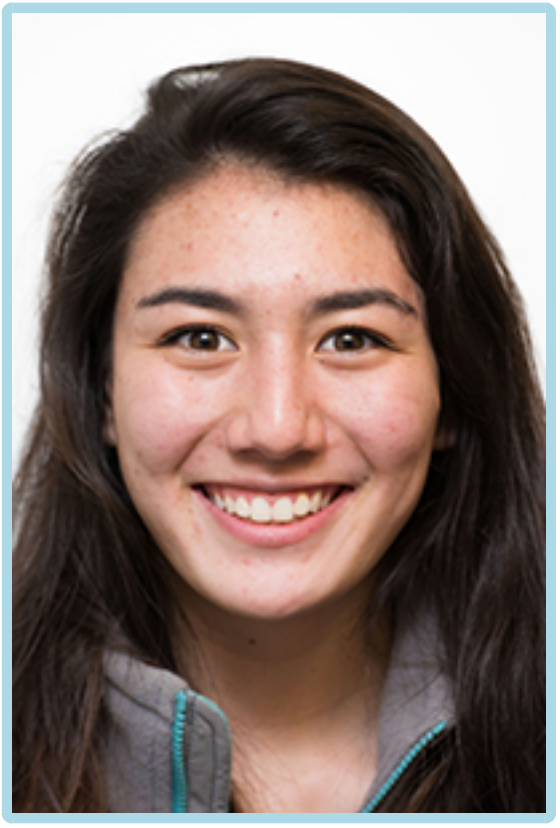
`erik@cs10.org`

Readers

Reader Jobel Vecino



Reader Katherine McGauley



Reader Lara McConnaughey

Grading

Grade Breakdown

For the most part, we would prefer to teach this course *without* grades. What a wonderful concept, learning for learning sake! However, even though we can't change the "system" overnight, we can create grading policies that support learning as much as possible. The various course activities will contribute to your grade as follows:

Activity	Points	Percent of Total Grade

Weekly Reading Quizzes	20	4%
Lab Check-Offs	30	6%
Homework 1 (Word Guessing)	10	2%
Homework 2 (Snowman)	20	4%
Homework 3 (Abominable Snowman)	30	6%
Midterm Project	75	15%
Innovation Blog	40	8%
Final Programming Project	75	15%
Quest	25	5%
Midterm	75	15%
Final Exam	100	20%
Total Points	**500**	**100%**

- For Reading Quizzes, each quiz will be worth 2 points, and your highest 10 scores will count.
- For Lab Check-Offs, each will be worth 2 points each, and your highest 15 scores will count.
- You may earn *"sprinkle points"* for up to 2 additional labs.

How We'll Calculate Your Grade

Your letter grade will be determined by total course points, as shown in the table below. Incomplete grades will be granted only for dire medical or personal emergencies that cause you to miss the final exam, and only if your work up to that point is satisfactory.

Points	Grade
485-500	A+
460-484	A
450-459	A-
440-449	B+
420-439	B
400-419	B-
375-399	C+
360-374	C
350-359	C-
300-349	D

- Point Ranges are inclusive.
- All final scores will be rounded to the nearest whole number.

Resources

- Blown to Bits
- Debugging Rules!
- UC Berkeley
- College of Engineering
- EECS Department Page
- (OLD) Solutions to Lab Exercises