

# Jerry Lung

jerry.lung@berkeley.edu | linkedin.com/in/jerrylung | (661) 678-3966  
jerrylung.com | github.com/jzlung

---

## University of California, Berkeley

B.A. Computer Science, May 2015

Technical GPA: 3.43

---

## IXL Learning

Aug 2015 - Present

### Software Engineer

- Redesigned our React training, reducing the onboarding of new hires to front-end work by 3 weeks
- Adapted quickly to resolve over 40 bug cases by my first release in 2 months
- Developed user-interactive components for the IXL main product using **Java**, **JavaScript**, and **React**
- Mentored 9 fellow new hires in their first front-end coding projects
- Pioneered new CommonJS codebase by rewriting key modules like Drag-and-Drop

## NICE Systems

June 2014 – Aug 2014

### Software Engineering Intern

- Boosted team's workflow by developing internal infrastructure tools to monitor application health levels and automate tasks (**Python**, **Bash**)
- Shipped a **Java** developer tool to assist with dependency management when updating software
- Adapted to an **Agile test-driven** environment using industry technologies such as **Jenkins** and **Splunk**

## UC Berkeley Electrical Engineering & Computer Science

Aug 2013 – May 2015

### CS162, CS61C Course Reader, CS61B Lab Assistant

- Developed for the grading infrastructure to manage hundreds of grades (see **Octobear** below)
- Facilitated students' understanding of Operating Systems, Data Structures, Algorithms
- Enhanced skills in understanding others' code through debugging C code in a class of 500 students

## PROJECTS

---

### Music Machine (HackJam 2014) - [musicmachine.herokuapp.com](https://musicmachine.herokuapp.com)

Project lead and developer on MEAN-stack web application for users to quickly query and download music (**Express.js**, **Node.js**, **MongoDB**).

### Octobear (Sp. 2015) – UC Berkeley EECS Instructional Staff

Team developer for new Linux grading infrastructure for EECS courses.

Contributed core algorithm for automatically parsing grade spreadsheets and entering into database.

Shipped interactive **Python** scripts for grade entering and syncing to **MongoDB** database, **Flask** webapp interface for students to monitor **Jenkins** autograder build queue and grades, CRON tasks

---

**Technical** Java, JavaScript, React, Python, Unix, HTML/CSS, Git + SVN, MongoDB + SQL,