Jerry Lung

www.jerrylung.com

jerry.lung@berkeley.edu
(661) 678-3966

LinkedIn www.linkedin.com/in/jerrylung
GitHub http://github.com/jzlung

University of California, Berkeley
B.A. Computer Science, May 2015
Technical GPA: 3.43

Technical Java, Python, Unix, HTML/CSS
JavaScript (Node.js, D3.js, jQuery)
SQL, Git, MongoDB, C++, C, Scala

EXPERIENCE

NICE Systems June 2014 – August 2014
Software Engineering Intern

- Worked as a software engineer with the Platform team to build internal infrastructure tools to boost development team's workflow and monitor application health levels (**Python, Bash**)
- Shipped a **Java** internal tool to assist with dependency management when updating stack software
- Adapted to an **Agile test-driven** environment and industry technologies such as **Jenkins** and **Splunk**

UC Berkeley Electrical Engineering & Computer Sciences August 2013 – Present
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

Berkeley Engineers and Mentors September 2012-Present
Webmaster, Curriculum, Site Leader

- Designed **http://beam.berkeley.edu** to be more informative and visually engaging
- Architected a 5-week high school project-based lesson to program Arduino remote control cars

PROJECTS

Music Machine (HackJam 2014) - **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. 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, other infrastructure tools.

Operating Systems & Distributed Computing

Implemented concurrency, multiprogramming, and system calls in OS and **Two-Phase Commit** logic in a Distributed Key-Value Store for database reliability; team project, modularity, unit-testing (**Java**)

Upper Level Coursework Product Development, Operating Systems, Databases, Security, Graphics, Networking, Artificial Intelligence, Architecture, Algorithms & Intractable Problems`