

# Davina Boedijono

6222 Encinita Ave, Temple City, CA 91780  
(626) 726-1449 | dboedijono@berkeley.edu  
www.linkedin.com/in/dboedijono

## EDUCATION

### Bachelor of Science in Electrical Engineering and Computer Science

Expected December 2016

University of California, Berkeley

GPA: 3.17

Completed Coursework: CS61A(Python), CS61B(Java and Data Structures), CS70(Discrete Math and Probability Theory), CS170 (Algorithms), EE20(Signals and Systems), EE40(Microelectronic Circuits)

Coursework In Progress: CS61C(Machine Structures), CS162(Operating Systems), CS168(Internet: Architecture and Protocols)

## TECHNICAL SKILLS

- Experienced in Java, C++, Python, C
- Proficient in web application development (HTML, CSS, Javascript, jQuery, Node.js), other programming languages (Scheme, Bash, Matlab, SQL), OpenGL, Qt Libraries, PyGame, LaTeX
- Comfortable working in Mac OS X, Unix, Windows environment with numerous development tools (Git, IDEs, debuggers)

## EXPERIENCE

### Computer Science Tutor | Self-paced Center, UC Berkeley

August 2015 – present

- Assisting students in numerous self-paced computer science courses, including Data Structures, C++, Java, Python, C, Scheme, and Unix in understanding the course materials and assignments (labs, homework, and projects)
- Grading the course assignments and going over past quizzes with each individual

### Software Engineering Intern | Intuit, Mountain View, CA

May 2015 – August 2015

- Adapted to Intuit Quickbooks Online (QBO) UI and QBO backend development environment, including the Java code base, Perforce, Git, Jenkins, and QuickBase
- Extended some QBO administrative servlets and designed a script in Bash to support the QA clusters testing in AWS
- Developed a single page web applications to put together QBO administrative servlets and other useful support tools that can help QBO developers, quality engineers, operations, and care agents access information and solve issues quickly
- Earned multiple recognitions of achievement from the Intuit software team for the usefulness of the software along with the resourcefulness, motivation, and independence demonstrated during the implementation of the assignment

### Head Reader CS70 (Discrete Mathematics and Probability Theory) | UC Berkeley

January 2015 – May 2015

- Guided students during the homework party and office hours, and graded students' homework and midterms
- Acted as the liaison between other course staff and the readers
- Coordinated the readers by creating a sign in sheet system and monitoring the readers' hours and attendance

### Lab and Office Hour Assistant CS61A | UC Berkeley

January 2015 – May 2015

Helped students in CS61A (The Structure and Interpretation of Computer Programs) with labs, homework, and projects

### Computer Science Tutor and Teaching Assistant | Pasadena City College

August 2013 – May 2014

- Assisted a professor in developing and grading exams and assignments for 3 classes and over 70 students
- Planned and facilitated a supplemental instruction program for CS students alongside 5 other CS faculty members
- Instructed supplemental lectures for more complex class topics, helping over 20 students at a time, which aided in a change of retention rate from around 50% of students dropping CS classes, down to 20% or less

## PROJECTS

### Gitlet Version Control System | Java

April 2015

- Designed and built a version control system in Java that supports numerous version control features from the simpler ones, like commit, merge, and rebase, to the more complicated features, such as remote and clone
- Utilized Java Serializable and built-in data structures, such as ArrayList, HashSet, HashMap, Stack, Queue to build an efficient version control system

### Type-racer Game | Python and PyGame

October 2014

- Participated in an 8-hour long Hackathon and worked with one teammate to create a typing speed game
- Engineered and designed a program that takes a text file, stores all the unique words in a randomized queue, and continually pops new words to the PyGame-based graphical user interface

### Book Reader Using Binary Tree | C++ and Qt Graphics Libraries

March 2014

- Collaborated with one other individual to build a program that parses a block of text, orders and counts individual words using an orchard of self-written balanced binary trees, then graphically displays results relevant to the text
- Communicated primarily online, however, we prioritized and worked efficiently to finish a project that greatly exceeded classroom expectations, beating the class average runtime of around 25 seconds with a runtime of 1.3 seconds
- Designed and programmed the data structures used, such as the binary tree, the parsing and word recognition algorithm, the processing required to give the user expected information, and the graphical user interface