Davina Boedijono

http://dboedijono.github.io/
www.linkedin.com/in/dboedijono/
(626) 726-1449 | dboedijono@gmail.com

EDUCATION

Bachelor of Science in Electrical Engineering and Computer Science **Expected December 2016**
University of California, Berkeley GPA: π
Completed Courses: CS61A(Python), CS61B(Java and Data Structures), CS61C(Machine Structures), CS170(Algorithms), CS162(Operating Systems), CS168(Internet Architecture and Protocols), CS186(Database Systems), CS161(Computer Security), CS164(Programming Languages & Compilers)

TECHNICAL SKILLS

- Fluent in Python, Java, C++; Experience in Django, Apache Thrift, C, PySpark, SQL and NoSQL databases
- Have worked with Android development and web application development (HTML, CSS, JavaScript, jQuery)
- Comfortable working in Linux, Unix, and a generous variety of development tools (Mercurial, Git, IDEs, debuggers)

EXPERIENCE

Software Engineering Intern | Instagram & Facebook, Menlo Park, CA **May 2016 – August 2016**

- Worked with Instagram Communication Infrastructure team developing Instagram Direct Message server and client
- Implemented heavy projects using Facebook's and Instagram's tech stacks including Python, Django, Apache Thrift, real-time systems, Java, Android, Mercurial, Git, and many other tools

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 Java servlets and designed scripts in Bash to support the QA clusters testing in AWS
- Developed a single page web application to conglomerate QBO administrative Java servlets and other support tools that 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

Computer Science Tutor | Self-paced Center, UC Berkeley **August 2015 – December 2016**

- Held 15+ hours of instructional and tutoring sessions weekly to help students understand the course materials and assignments for numerous self-paced computer science courses, such as Data Structures, C++, Java, Python, and C
- Graded the course assignments (quizzes, homework, and projects) and going over past quizzes with each individual

Head Reader CS70 (Discrete Mathematics and Probability Theory) | UC Berkeley **January 2015 – May 2015**

- Guided students during homework parties 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, and Queue to build an efficient version control system

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