Christopher Meredith

Software Engineer

Contact

E-mail: chrismeredith@berkeley.edu
Website: christophermeredith.com

Profile

Areas of particular interest to me include networking, network and internet security, algorithms, and language design. I also enjoy working on projects where I can make an impact on the end product and end user experience.

Work Experience

Google — **Irvine, CA**
Software Engineer — **January 2013 - Present**

- Frontend and backend development for the in-product help system.
- Expose relevant help content and contact options based on user context.
- Languages used: Java, JavaScript (AngularJS, Closure)

Qualcomm — **San Diego, CA**
Software Engineering Intern — **May - August 2012**

- Modem-related driver development for Windows 8 phone and tablet using Microsoft's Kernel-Mode Driver Framework.
- Designed a framework for remote testing of the modem over a network.
- Languages used: C, Python

Informatica — **Redwood City, CA**
Software Engineering Intern — **May - August 2011**

- Data modeling, data conversion, and thread management for the workflow feature.
- Integrated workflow feature with an external engine for state management.
- Language used: Java

Programming Languages

C/C++	Python	JavaScript
Java	Ruby/Rails	Lua

Education

University of California, Berkeley
Bachelor of Science
Electrical Engineering & Computer Sciences (2012)
3.605 GPA

Relevant Courses

Data Structures	Languages and Compilers	Security
Machine Structures	Artificial Intelligence	Networking
Algorithms	Databases	Graduate Network Security
Operating Systems	Graphics	Software Engineering

Course Projects

Fall 2012: Worked with four other students to engineer a web app for the Berkeley Community Fund – a non-profit organization that provides college scholarships for high school seniors. We used Ruby on Rails to replace their paper application process with an online system. The app also supports recommendations, interview scheduling, mentor assignment, and generation of statistics.

Spring 2012: Worked with a partner to implement a bytecode interpreter for a subset of Lua (supporting coroutines), a parser, and a compiler-compiler. We then defined a grammar for a subset of HTML (with parser actions for creating DOM objects) and implemented scripting and event stream constructs for dynamic manipulation of the DOM objects.

Fall 2011: Worked with four other students to implement a thread-safe (using our own synchronization primitives) Client/Server for the game of Go. The server can handle multiple games concurrently and uses a SQLite database to store player authentication information, track game history, and recover from failures. This was all deployed and tested on Amazon EC2 virtual machines.