

Samuel C. Pullman

2425 Warring Street, Berkeley, CA 94704 • (619) 248-4018 • sampullman@berkeley.edu

- Recent UC Berkeley graduate looking to apply technical knowledge and class instruction in the field of Computer Science through a full time position in the industry.
- Interested in many aspects of software design, development, and testing, from system programming to mobile application development.
- Ability to quickly learn and apply new concepts, and adapt to new settings; proficient in numerous programming languages, tools, testing methodologies, and version control systems.

Education

University of California Berkeley
B.S. in Electrical Engineering and Computer Science

Berkeley, CA, Degree received May 2012

- Overall GPA: 3.4

Work Experience

Undergraduate Student Instructor (UGSI) for CS61A (January 2012 – May 2012)

- Worked for one semester as a UGSI for an introductory computer science course at UC Berkeley.
- Taught three hours of discussion section and three hours of lab per week.
- Assisted with writing labs, discussion notes, and exam questions, held office hour sessions twice a week.

Programmer and Web Developer – Lawrence Berkeley National Lab (LBNL) (June 2010 to January 2011)

- Assisted in the development of a web based interface for a self teaching expert system (SeTES).
- Project that will significantly improve the effectiveness and efficiency of analyzing, predicting, and designing natural gas production from extremely tight unconventional gas reservoirs.

SULI (Science Undergraduate Laboratory Internships) (Summer 2009)

U.S. Department of Energy Berkeley, CA 2009

- Worked with LBNL researcher Dr. Jinsong Chen writing code to statistically analyze inverted geological data and developed a web based interface for Dr. Chen's stochastic inversion codes
- Abstract: adsabs.harvard.edu/abs/2009AGUFMNS31B1168C

Qualifications

- Experienced programmer in C, C++, Python, Java, Visual Basic 10, Javascript, JQuery, PHP, Lisp (Scheme), and MIPS
- Experience developing Android, Symfony, and Ruby on Rails applications
- Knowledge of Unix, Linux (Debian), Windows, and Macintosh systems
- Able to collaborate well on projects through version control systems such as Subversion and Git

Relevant Coursework

Computer Science 162 – Operating Systems and Systems Programming

- Introduction and applications to processes, interprocess communication, and synchronization.
- Techniques for resource allocation, scheduling, performance evaluation, along with file systems, memory allocation, and security.

Computer Science 164 – Programming Languages and Compilers

- The design of modern programming languages.
- Principles and techniques of scanning, parsing, semantic analysis, and code generation. Implementation of compilers, interpreters, and assemblers. Overview of run-time organization and error handling.

Computer Science 170 – Efficient Algorithms and Intractable Problems

- Concept and basic techniques in the design and analysis of algorithms; models of computation; lower bounds.
- Turing machines, how to count steps, deterministic and nondeterministic Turing machines, NP-completeness.

Computer Science 186 – Introduction to Database Systems

- Hierarchical, network, relational, and object-oriented data models.
- High-level interfaces including application generators, browsers, and report writers.

Computer Science 188 – Introduction to Artificial Intelligence

- Basic ideas and techniques underlying the design of intelligent computer systems
- Class projects based in python language, extensive knowledge gained of the language through consistent use.

Computer Science 184 – Computer Graphics

- Techniques of modeling objects for the purpose of computer rendering: boundary representations, constructive solids geometry, hierarchical scene descriptions.
- Algorithms for clipping, hidden surface removal, and anti-aliasing, along with scan-line and ray-based rendering algorithms.

Computer Science 161 – Computer Security

- Cryptography, including encryption, authentication, hash functions, cryptographic protocols, and applications.
- Operating system security, access control, with case studies from real-world systems.

Electrical Engineering 122 – Introduction to Communication Networks

- Introduction to the design and implementation of computer networks, focusing on the concepts and fundamental design principles that have contributed to the Internet's scalability and robustness.
- Topics include layering, congestion/flow/error control, routing, addressing, multicast, packet scheduling, switching, internetworking, network security, and networking/programming interfaces.

Electrical Engineering 120 – Signals and Systems

- Continuous and discrete-time transform analysis techniques. Fourier series, Fourier transform, Laplace and Z-transforms, Sampling and reconstruction.
- Solution of differential and difference equations using transforms.

Other Experience

- | | |
|---|---------------|
| • Lab Assistant - Computer Science 3L | (Spring 2009) |
| • Lab Assistant - Computer Science 61bl | (Summer 2010) |
| • Treasurer for the California chapter of Delta Upsilon | (2010) |
| • Webmaster for Delta Upsilon | (Fall 2009) |
-