Brian Zhang

Software Engineer.

925-699-3883	Berkeley, California	brian.zhang@berkeley.edu
PHONE	**LOCATION**	**EMAIL**

PROFILE

Computer science graduate from the University of California, Berkeley. Interests include web development, programming languages, and machine learning. In my free time I enjoy spending time outdoors and playing jazz piano.

SKILLS

Web Frameworks

AngularJS, Express, Node.js, React, Django, Flask

Programming Languages

C, C++, CoffeeScript, Objective-C, Java, JavaScript, Python, Ruby, Scala, Swift

Operating Systems

Arch, Debian, Fedora, OS X, Windows

EXPERIENCE

Software Engineering Intern, Symmetricom, Inc. May 2013–Aug. 2013

Source control management system administration.

- Modernized source control management system from Mercurial to Git and Atlassian Bitbucket/JIRA.
- Worked on new software development workflow, e.g. automating bug reporting and tracking services.

Software Engineering Intern, Blackhawk Network May 2012–Aug. 2012

Front-end and back-end financial services.

- Developed iOS app for GoWallet, a service for managing financial payments products, e.g. gift cards.
- Developed back-end payment processing system for PayPower, a reloadable Visa prepaid card.

EDUCATION

University of California, Berkeley Jan. 2011–Dec. 2015

Bachelor of Arts, Computer Science

- Artificial Intelligence, Computer Security, Database Systems, Efficient Algorithms and Intractable Problems, Operating Systems and System Programming, Programming Languages and Compilers

PROJECTS

Logo Interpreter (Python)

Implemented evaluator, parser, and procedures for the Logo language.

LIFC Compiler (C)

Compiler from the homebrew language of LIFC to MIPS assembly; implemented lexing, parsing, static analysis, and code generation.

Arbor (JavaScript)

Small domain-specific language designed to visualize abstract syntax trees using the D3 JavaScript library; developed in two weeks by my partner and myself.

NP-Hard Approximation (Python)

Approximation algorithm for maximum acyclic subgraph problem proven to yield a better than 1/2-approximation on a majority of instances.

Operating Systems Term Project (C)

Implemented processes, memory allocation, resource allocation, scheduling, and user programs; developed with Pintos, a simple operating system framework.

- Features alarm clock, priority scheduler, multilevel feedback queue scheduler, and distributed key-value store database with two-phase commit.

Programming Languages Term Project (JavaScript)

Simple language called CS164 inspired by Lua; implemented interpreter, objects, layout engine, compiler from ASTs to bytecode, bytecode interpreter, and grammars; developed with Node.js.

- Features integer arithmetic, closures, desugaring, objects with prototype-based inheritance, coroutines, and tail call optimization.

Database Systems Term Project (Java/Scala)

Simple database system; implemented query joins, query optimizer, and recovery; developed with Amazon SimpleDB and Apache Spark.

- Features chunk-nested loop join, symmetric hash join, dynamic programming and greedy join algorithms, log-based rollback, disk hash-partitioning, in-memory UDF caching, and hash-partitioned UDF caching.

Artificial Intelligence Term Project (Python)

Designed agents for the classic Pacman game; implemented multi-agent search, reinforcement learning, inference algorithms for Bayes' nets, and classification.

- Search features depth-first, breadth-first, uniform-cost, A* search, minimax with alpha-beta pruning, and expectimax agent.
- Learning and inference features value iteration, Q-learning, variable elimination, likelihood weighting sampling, hidden Markov models, and particle filtering.
- Classification features perceptron, large-margin (MIRA) classifier, and perceptron classifier for behavioral cloning.

REFERENCES

Available upon request.