Kyle Chia-Ching Chang

2 Mountain Laurel, Irvine, CA 92604 | C:(949)923-9997 | kcc020@ucsd.edu

Education

University of California, San Diego
B.S. Mathematics-Computer Science
[CSE GPA: 3.6] | [Expected Graduation Date: 06/2019]

- Excelled in Advanced Data Structures, Design and Analysis of Algorithms, SPARC Architectures and Assembly, Discrete Mathematics, Vector Calculus, Linear Algebra, Differential Equations

Skills

- **Programming Languages**: C++, C, SPARC Assembly, Java, Python
- **Web Technologies**: HTML, CSS, Javascript, Heroku, Git
- **Development Tools**: VIM, Valgrind, GDB,
- **Languages**: English, Mandarin (AP Chinese)

Experience

Computer Science Tutor (September 2016-December 2016)(January 2016-Current)

- Tutored students in CSE 8A ("Introduction to Programming Java I), a class of over 400 students.
- Advised and taught beginner students object-oriented programming, fundamental concepts in computer science, and building/designing programming constructs and applications.

Projects

Snake (Java)

- Created the classic Snake video game in Java that could respond to key inputs.
- Worked with mouse and key listeners, implemented GUI interfaces, and used polymorphism.

Personal Website (HTML, CSS, Javascript)

- Created a personal website to display resume, current projects, etc.
- Learned and applied standard, web development conventions.

ls Unix command (C, SPARC Assembly)

- Created an implementation of the ls Unix command.
- Learned to filter, prompt, and stores user input.

Autocomplete program (C++)

- Worked with a partner to create a program that could "autocomplete" a string based on the first n characters entered.
- Used a TST data structure to store most popular words so the autocomplete would return the most likely started word.

Huffman Tree (C++)

- Worked with a partner to create a loss-less compression program.
- Used the Huffman compression algorithm to ensure the smallest compression without losing any data or accuracy.

Six Degrees of Kevin Bacon Graph Project (C++)

- Worked with a partner to create a program to find the shortest connections between actors with mutual movies acting as edges.
- Used and implemented BFS, Dijkstra's algorithm to find the shortest connections.