Benjamin Huang

Dublin, CA (510) 509-0725 bejhuang@gmail.com

Education

University of California, Berkeley December 2022
Computer Science

Relevant Coursework: Data Structures, Algorithms, Computer Architecture, Computer Security, Advanced Programming in C++, Discrete Mathematics and Probability Theory, Advanced Linear Algebra

Skills

Programming Languages:	Python, Java, C++, C, Go, SQL
Technical:	Data Structures, Algorithms, Object-Oriented Programming, Linux

Projects

Scheme Interpreter *Python*

- Developed an interpreter for a subset of the Scheme programming language (iteration of Lisp)
- Incorporates assignment statements, functions, lambda/higher-order functions, and self-evaluating expressions
- Models the current state of the program using environment frames which keep track of variables and functions in their corresponding scope

Penetration Testing *SQL, Javascript*

- Course project involving a website intentionally designed to be vulnerable to various cyber attacks
- Exploited backend SQL code by using SQL injection attacks to reveal user information, granting access into the website as a specific user
- Performed stored and reflected XSS attacks to run malicious Javascript on a user which then leaked their session cookie or deleted all their files on the server

Typing Speed Test *Python*

- Generates a sentence for a user to type which then calculates their WPM, accuracy, and time
- Features a recursive autocorrect algorithm that intelligently finds a word most closely matching a mistyped word
- Implemented multiplayer functionality which tracks progress between two players in real time by sending information to a multiplayer server

CPU Design *Logisim*

- Created a CPU using built-in tools from Logisim in order to be able to perform the five stages of executing a function: Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory (MEM), Write Back (WB)
- Built the control-logic in order to decode a vast set of instructions which models the flow of execution of a physical CPU
- Implemented pipelining which optimized the rate at which work can be done by having instructions be executed in parallel