# Daniel Van Der Maden

dvandermaden0@berkeley.edu (714) 251-4974
danielvandermaden.com github.com/Daniel-VDM linkedin.com/in/Daniel-VDM

## EDUCATION

**University of California, Berkeley** **Berkeley, CA**
*B.A. Computer Science* – GPA: 3.68/4.00 *December 2019 (Expected)*
- *Areas of academic interest*: Human-Computer Interaction, Natural Language Processing, and Artificial Intelligence.
- *Relevant coursework*: Database Systems (IP), UI/UX Design (IP), Computer Security (IP), Efficient Algorithms, Intro to Artificial Intelligence, Natural Language Processing, Intro to Linguistics, Machine Structures, Data Structures, Structure & Interpretation of Programs, Designing Information Devices, Discrete Math & Probability Theory, Linear Algebra & Differential Equations.

## EXPERIENCE

**Microsemi** **Aliso Viejo, CA**
*Software Engineering Intern* – Frequency and Timing Division *May 2018 – August 2018*
- Gained practical industry experience by developing embedded programs for a modernized eLoran (radio navigation and data) system.
- Prepared the project for live tests by implementing multiple signal schemes and interfacing them with a custom transmission timer.
- Improved the custom transmission timers by extending the API to allow for more versatility in the signal scheme implementation.
- Shortened scheme implementation time by creating data collection and development tools as well as writing thorough documentation.

**EECS Department, UC Berkeley** **Berkeley, CA**
*Tutor* – Self-Paced Center *January 2019 – May 2019*
- Improved student understanding by holding office hours for students learning C, C++, Java, and Python.
- Graded projects and gave feedback that encouraged industry-ready practices. Also graded quizzes and final exams.

*Academic Intern* – Structure and Interpretation of Computer Programs Course *August 2018 – December 2018*
- Fostered interest in Computer Science fundamentals by providing guidance on homeworks and projects during office hours.
- Assisted course staff during lab sessions by conducting lab checkoffs and answering questions regarding course content.

**Computer Science Mentors, UC Berkeley** **Berkeley, CA**
*Mentor* – Designing Information Devices and Systems I Course *January 2019 – May 2019*
- Helped students solidify core EECS concepts by leading a supplemental 1.5 hour discussion section each week.
- Mentored a cohort of students throughout the semester and helped develop supplemental worksheets and lesson plans.

**J-Sei Community Center** **Emeryville, CA**
*Technology Consultant for Seniors* *September 2017 – December 2017*
- Effectively communicated with the elderly a verity of tech safety topics such as internet privacy and malware protection.

## NOTABLE PROJECTS

*A detailed list (with repo links) can be found on my personal website*

**Easy Seq2Seq Chatbot** – Personal Project Link to repo
- A chatbot that is implemented using a sequence to sequence model with an easy way to define training and model parameters.
- It features a caching system, saving and loading models, interrupt recovery, memory efficient training, and multiple data filters.

**Approximate Solver for a NP-Hard Problem** – Efficient Algorithms Course Project Link to repo
- A polynomial time approximate solver that uses a greedy algorithm with various heuristics (each is considered, best is chosen).
- Yielded solutions that were in the top 10% of all approximations in the course (which had $\approx$ 700 students).

**Personal Website** – Personal Project Link to repo
- A mobile and ultra-wide friendly website that is created using Bootstrap and some JavaScript with jQuary.

**Coreference Annotator** – Natural Language Processing Course Project Link to repo
- Annotates the antecedent for each pronoun in a given dataset using a logistic regression model. Achieved 71% accuracy on test data.

**Concurrent Cached File Server** – Machine Structures Course Project Link to repo
- A file server (written in Golang) with a cache that can efficiently handle thousands of concurrent file requests (on a laptop).

**61Ccc Compiler (to RISC-V)** – Machine Structures Course Project Link to repo upon request
- Compiles code that is written in the 61Ccc language (a made-up subset of C) into universally compatible RISC-V assembly.

## SKILLS

**Programming Languages:** Python, Java, C++, C, Golang, Scheme, RISC-V/x86, SQL, HTML, CSS, JavaScript, LaTeX
**Technologies:** Scipy/Numpy/Pandas, NLTK, Keras, Scikit-learn, Hadoop, OpenMP, Intel AVX, Bootstrap, MySQL, AWS EC2
**Spoken Languages:** Native English speaker, fluent in French, basic understanding of Vietnamese.