

# Devin Lehmacher

112 Sage Place Room-B09, Ithaca, NY 14850  
djl329@cornell.edu • +1 (864) 722-3014 • [github.com/lehmacdj](https://github.com/lehmacdj)

## EDUCATION

**Cornell University**, Ithaca, NY 14853

Aug 2015 — Present

- Expect to graduate May 2019
- Cumulative GPA: 3.45
- Bachelors of Arts in Computer Science (Major GPA: 3.82)
- Bachelors of Arts in Biology (Major GPA: 3.60)

## CLASSES

**Advanced Programming Languages**, CS 6110

Spring 2017

**Operating Systems & Practicum**, CS 4410 & CS 4411

Spring 2017

**Database Systems**, CS 4320

Fall 2016

**Computer System Organization**, CS 3410

Fall 2016

**Functional Programming and Data Structures**, CS 3110

Spring 2016

**Discrete Structures**, CS 2800

Spring 2016

**Object Oriented Programming and Data Structures**, CS 2110

Fall 2015

**C++ Programming**, CS 2024

Fall 2016

## WORK EXPERIENCE

**Teaching Assistant**, CS 2110 at Cornell University

Spring 2016 — Present

- Teach a section with about 25 students each week
- Weekly office hours to help students understand the course material
- Weekly meeting to discuss course material and plan for the next week
- Grade assignments to give students helpful feedback for next assignment

**Research Assistant** at Clemson University

Jun 2015 — Aug 2016

- Tested the performance of MedusaLoop, a program that models protein loops
- Analyzed the results of tests creating figures to show visualize bad performance
- Wrote a daemon to dispatch jobs from a database to a server instance
- Wrote back end code that interacted with a database to fetch and write new jobs

## PROJECTS

**OCalf Interpreter**, CS 3110

- Built an interpreter for a small subset of OCaml
- Learned how to evaluate an AST for a functional language using small step semantics
- Implemented Hindley-Milner type system and inference algorithm to type check OCalf programs

**Scheme Interpreter**, [github.com/lehmacdj/haskell\\_scheme](https://github.com/lehmacdj/haskell_scheme)

- Built an interpreter for a relatively large subset of Scheme
- Learned how to implement the semantics for dynamically typed programming languages
- Learned how to build a parser using Parsec

**Heaplib**, CS 3410

- Implemented and tested malloc, free, and resize in C
- Learned how to use raw pointers and the trade-offs involved with building an allocator
- Wrote a large number of tests to ensure that pointer arithmetic was correct

**MIPS Processor**, CS 3410

- Designed a MIPS processor in Logisim and tested it with programs written in assembly
- Learned how to decode binary MIPS instructions into control signals
- Gained understanding about how processors execute instructions

**OCaml Ed**, [github.com/lehmacdj/ocaml-ed](https://github.com/lehmacdj/ocaml-ed)

- Implementation of ed, the 1960s line editor, written using OCaml
- Wrote clear error handling code that cleanly passes errors up to the top level
- Learned how to independently design a large project and document modules effectively

**Life Simulator**, [github.com/lehmacdj/simulation](https://github.com/lehmacdj/simulation)

- Implemented the Game of Life and multicolor variants using Rust
- Learned how to write memory safe code using Rust and generate png images

## SKILLS

- Java (3 years), Vim (2.5 years), Shell (2.5 years), git (2 years), OCaml (1 year)
- Haskell (1 year), C (1 year), SQL (1 semester), Rust (1 semester), C++ (1 semester)