

# JASPER GEER

jasper.geer@gmail.com | www.jaspergeer.com

## EDUCATION

---

### Tufts University

Medford, MA

Computer Science Major

2020-2024 (*Expected*)

Major GPA: 3.89

GPA: 3.86

- Coursework: Compilers, Virtual Machines and Language Translation, Programming Languages, Graph Theory, Advanced Topics in Computer Architecture, Internet-Scale Distributed Systems, Operating Systems
- Activities: Tufts Chinese Students' Association Event Chair 2022-23

### Mercer Island High School

Mercer Island, WA

2016-2020

## RESEARCH EXPERIENCES

---

### Tufts Programming Languages (TuPL), Tufts University

Research Assistant

September 2023 - Present

- Conducted program synthesis research under Professor Jeff Foster.
- Worked on the implementation of a novel constraint-guided Java program synthesis technique.

### Tufts Security and Privacy Lab, Tufts University

Research Assistant

September 2023 - Present

- Assisted in a review of recent symbolic execution literature under Professor Dan Votipka.
- Qualitatively coded rounds of 5-10 research papers and contributed to codebook development.

## PROFESSIONAL EXPERIENCES

---

### Tesla

Vehicle Software Intern

May 2023 - August 2023

- End-to-end feature development in Haskell for an incremental compiler frontend.
- Refactored compiler passes into incremental build rules for a monadic build system.
- Created embedded domain-specific languages to implement new language server features.
- Received offer for full-time conversion.

## TEACHING

---

### Tufts University, Teaching Assistant

September 2022 - Present

- CS170, Computation Theory. *Fall 2022*.
- CS170, Computation Theory. *Spring 2023*.
- CS170, Computation Theory. *Fall 2023*.

### Coding With Kids

May 2022 - September 2022

- Taught week-long programming classes for middle school and elementary school students

## AWARDS

---

- Travel Award: Programming Languages Mentoring Workshop (PLMW) at International Conference on Functional Programming (ICFP) 2023

### Compost

- LLVM frontend for a statically-typed functional programming language.
- Designed an affine type system to enforce memory safety without runtime garbage collection.
- Began as a personal summer project, completed as a semester-long group project in a compilers class.
- Written in OCaml.

### tinyvalidator

- Artifact produced for directed study with Professor Jeff Foster.
- Translation validation for a C-subset language by means of symbolic execution.
- Devised a big-step operational semantics to describe the execution of programs with symbolic inputs.
- Written in Haskell using the Z3 SMT solver.

### Odin

#### *Contributor*

- Contributed bug fixes to the open-source compiler for the Odin programming language.
- Addressed fatal type checker errors that occurred for certain well-typed programs.
- Diagnosed and fixed code generation bugs triggered by casts and non-native endianness.

## PROGRAMMING BACKGROUND

---

- Recent Experience with Haskell, OCaml, Scala, Agda, and C.
- Some experience with SML, Scheme, C++, Typescript, Python, and Java.