# JASPER GEER

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

## Education

**Tufts University** *2020-2024 (Expected)*
*Medford, MA* Major GPA: 3.89
Computer Science Major GPA: 3.86

**Mercer Island High School** *2016-2020*
*Mercer Island, WA*

## Research Experiences

**Tufts Programming Languages (TuPL)**, Tufts University *September 2023 - Present*
*Research Assistant*

- 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 *September 2023 - Present*
*Research Assistant*

- Assisted in a review of recent symbolic execution literature under Professor Dan Votipka.
- Qualitatively coded weekly rounds of 5-10 research papers and experience reports.

## Professional Experiences

**Tesla** *May 2023 - August 2023*
*Vehicle Software Intern*

- 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

## Projects

**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.