

# Lucas Sta Maria

lucas.stamaria@gmail.com | +1(857) 800-2586 | GitHub: [priime0](#) | Website: <https://priime.dev> | [linkedin.com/in/lucas-sta-maria](https://www.linkedin.com/in/lucas-sta-maria)

## EDUCATION

### Northeastern University

Boston, MA

Bachelor of Science in Computer Science and Mathematics

Expected May 2025

**Relevant Courses:** Programming Languages, Compilers, Algorithms, Advanced Probability and Statistics, Graph Theory

**Teaching Assistant:** Programming Languages (Spring 2023, Fall 2023)

## EXPERIENCE

### TensorDock

August 2023 – Present

Software Engineer Intern

Cambridge, MA

- Architected and implemented an asynchronous API wrapper client in **Rust** and **Tokio** to interact with the Marketplace API.
- Developing a command-line utility in Rust to facilitate fast deployment of Marketplace cloud GPUs and manage instances.
- Maintaining the **Python** and **Flask** Marketplace backend, reducing user-facing bugs when deploying GPU servers.
- Introduced stricter client and server-side validation, along with fine-grained permissions-scoping for auth token generation.

### Amazon

May 2023 – August 2023

Software Development Engineer Intern

Seattle, WA

- Architected and introduced five new passes to an internal compiler in **Java** to transform between two critical Alexa developer configuration formats, helping backfill **500+** configuration files and improving Alexa developer velocity.
- Designed a new intermediate representation (IR) to support metadata labels, external schema files, and composed attributes.
- Refactored existing architecture of the compiler to support multiple stages of different IRs by using the visitor pattern.
- Wrote a code generation pass for the compiler that recursively traversed the IR to generate its corresponding configuration.
- Developed a comprehensive suite of unit and end-to-end tests with **JUnit**, maintaining **95+**% test coverage.

### Cigna

May 2022 – August 2022

Software Engineer Intern

Boston, MA

- Optimized performance throughput by **40%** for an existing insurance claim eligibility microservice receiving **100,000+** daily requests by improving the **SQL** database queries.
- Added endpoints to support an organization-wide version 2 of the API, accommodating new information while maintaining backwards-compatibility with **C#** and **.NET**.
- Increased responsibility of the microservice by adding functionality for analyzing and organizing claims, then redirecting them to their respective microservices.

## PROJECTS

### x64 Compiler | OCaml, C, x64

January 2023 – April 2023

Functional Programming Language

- Integrated A-Normal Form IR for non-trivial expression code generation to **x64 assembly** in **OCaml** with a **C runtime**.
- Implemented garbage collection with Cheney's semispace collector algorithm by walking the call stack, marking and copying used allocations to compact used memory and remove fragmentation.
- Introduced phases for optimized register allocation by producing a colored graph by static analysis on variable liveness.
- Added static bidirectional type-checking and inference by constraint-solving on the IR types with **Racket** and **miniKanren**.

### UFDS Training / Rhuver | Rust, TensorDock, React.js, Evergreen, Firebase

June 2021 – Present

Competitive Programming Training Platform

[Website](#)

- Scaled the platform with **Firebase** to help over **1000 active users** achieve top placements in competitions (IOI, APIO, INOI).
- Migrated code evaluation service dependency from AWS Lambda to TensorDock, reducing operational costs by **90%** and achieving newly defined SLAs of executing 100 complex tasks within 3 seconds.
- Refactored Rust codebase to dispatch and containerize jobs with IOI Isolate.

## AWARDS

**Hackathon Awards:** Hack@Brown 2023, HackBeanpot 2023, NUS Hack&Roll 2021, OpenHacks 2020

**AWS Award, Shortlist (Top 10)**, International Space Challenge

January 2021

**Bronze Medal**, National Olympiad in Informatics, Singapore

April 2020

## SKILLS

**Programming Languages:** Python, Java, JavaScript, Rust, Lisp (Racket, Elisp, SBCL), OCaml, C++, SQL

**Tools & Technologies:** Git, Bash, Linux, AWS (Lambda, S3), PostgreSQL, Redis, SQLite, Docker, GitHub Actions