Remus Trapsa

Cambridge, ON | rtrapsa@uoguelph.ca | LinkedIn | GitHub | (226) 789-8096

EDUCATION

University of Guelph - Guelph, ON
Bachelor of Computing, Computer Science (Co-op) | Expected Graduation: April 2030

- Cumulative GPA: **3.7/4.0**
- Relevant areas: Object-Oriented Programming, Software Development, Data Structures, Algorithms, Databases

Guelph Collegiate Vocational Institute - Guelph, ON
Ontario Secondary School Diploma, IB Diploma - Class of 2025

- Ontario Scholar Award
- Top 25% in Fermat Math Contest

TECHNICAL SKILLS

Programming: Python, Java, JavaScript, TypeScript, Go, C
Web Development: HTML, CSS, SCSS, React, Next.js, Chart.js
Databases, Cloud & APIs: PostgreSQL (Neon), Google Cloud Run, Claude API, GitHub Webhooks (HMAC)
Version Control & Environment: Git, GitHub, Linux/Unix command line
Concepts: Object-Oriented Programming, Application Development, Debugging, Input Validation, Automated Testing, Database-Backed Applications

TECHNICAL PROJECTS

ResolvePR - AI Code Security Agent | May 2026
Go, Claude API, Google Cloud Run, PostgreSQL (Neon), Next.js, Vercel

- Developed a Go engine that analyzes pull-request changes at the syntax-tree level to identify potential security issues in modified functions.
- Designed and implemented an automated testing approach and API guardrails, tuning confidence thresholds to reduce false-positive security findings by approximately **40%**.
- Developed an HMAC-verified GitHub webhook pipeline on Google Cloud Run with PostgreSQL (Neon), paired with a Next.js/Vercel dashboard for severity filtering, false-positive dismissal, and real-time critical and high-severity alerts.
- Collaborated with a team of four to design, test, troubleshoot, and deliver the complete application during a 24-hour hackathon.

Bolt Length Calculator | January 2025 – March 2025
React, TypeScript, Chart.js, SCSS

- Built a React/TypeScript app that computes required bolt length based on component thicknesses, washer stack-up, and bolt diameter, rounding up to the nearest standard 0.25" increment.
- Persisted calculation history to localStorage with delete and undo functionality, and graphed length trends over time with Chart.js.
- Designed a responsive Materialize CSS/SCSS interface with input validation to guide users through multi-field calculations.

Object-Oriented Chess Engine | November 2024 – January 2025
Java

- Collaborated with two peers to build a fully playable chess application in Java using object-oriented design principles
- Designed modular classes for chess pieces and game components to improve maintainability and reuse.
- Implemented and debugged core application logic including piece movement, turn management, game-state validation, and rule enforcement.
- Tested game behavior and resolved logic errors across different gameplay scenarios.