

# RANDY ZHU

604-704-9500 | [randy@randyzhu.com](mailto:randy@randyzhu.com) | [linkedin.com/in/rzhuo8](https://linkedin.com/in/rzhuo8) | [randyzhu.com](https://randyzhu.com) | [github.com/RandoNandoz](https://github.com/RandoNandoz)

## EDUCATION

### University of British Columbia

Bachelor of Science, Honours Computer Science, Option in Software Engineering

September 2023 – December 2027

GPA: 88%

## TECHNICAL SKILLS

**Languages:** Python, Java, C#, TypeScript, HTML/CSS, C++, C, SQL, MIPS Assembly, Swift

**Developer Tools:** Git, Docker, Linux

**Testing Frameworks:** JUnit, NUnit, pytest, Playwright

**Technologies:** React.js, Google Maps API, Unity Game Engine, Express.js, Google Cloud Run, Power BI, SQL Server Management Studio (SSMS), pandas, pthreads

## WORK EXPERIENCE

### Research Assistant

May 2025 – September 2025

Software Practices Lab

Vancouver, BC

- Implemented **graph algorithms for static and dynamic program analysis in Python** for automated unit test generation
- Identified external dependencies to functions by extracting docstrings from objects for analysis with an open source LLM, **Gemma 3n on ollama**
- Developed the tool using Agile methodologies like **Kanban**; managed source code collaboration using **Git**
- Caught 76% of bugs and covered 85% of code base using test-driven-design by writing over 300 **unit tests in pytest**, **mocking expensive API calls using pytest\_mock and monkeypatch**

### Software Developer Intern

September 2024 – April 2025

Teck Resources

Vancouver, BC

- Created calendar component used by **teams across the org** using the Power Apps Component API, React.js, TypeScript and the Microsoft Fluent UI React toolkit
- Tested web apps, catching 87% of bugs before reaching user acceptance tests using **Playwright** and **NUnit** and **C#**
- Saved over 100 hours for site engineering teams by creating a data ingest tool using the Microsoft Dataverse **REST API** in **C#**
- Presented **Power BI** dashboard of on-site safety events across business units by unifying data from legacy databases, using **SSMS**, **to wrangle data using SQL**, then finer transformations using **M**, **DAX** and **pandas** for the final dashboard

### Teaching Assistant

July 2024 – Present

The University of British Columbia

Vancouver, BC

- Achieved a 98% favourable rating from students for **debugging their event-driven Swing code in Java**
- Lectured during seminars on low-level programming fundamentals like **MIPS Assembly**, **stack frames**, and **POSIX pthreads**
- Explained and solved parallel programming problems as a part of office hours using **spinlocks**, **mutexes**, and **threading**

## PROJECTS

### Racket Compiler | x86 assembly, Linux ABI, C, Racket (Scheme/Lisp), gdb

2026

- Implemented **instruction selection and register allocation using graph colouring algorithms** to minimize expensive memory reads
- Working towards support for first class functions via closure conversion, and **tail call optimization to enable efficient recursion**

### Campus Explorer | TypeScript, React.js, express.js, Google Maps API, Docker, Google Cloud Run

2025

- Designed REST APIs and implemented them in express.js, writing middleware to route rooms using the Google Maps Routing API for cycling and walking paths
- Containerized the backend using **Docker** and **deployed the API to Google Cloud Run**
- Displayed and transformed embedded map using the Google Maps JavaScript API on React.js based frontend

### iscsi-driverkit | Swift, C++, DriverKit, Networking, iSCSI (RFC 7143)

2025

- iSCSI is a storage protocol similar to SMB. macOS has no support for it, and third-party addons use deprecated kernel modules and are closed source. This project aims to create a GPLv2, trusted, open-source implementation that uses modern APIs (i.e., no kernel extensions).
- Working around limitations of DriverKit by **implementing the networking client in user space with Swift**
- Implemented SCSI device driver with ring buffers to improve performance using **C++** without the STL