

Bryce Hart

Bryce.Hart23 [at] gmail [dot] com | LinkedIn: bryce-hart1 | GitHub: Bryce-Hart1

EDUCATION

East Carolina University

Bachelor of Science in Computer Science

Greenville, NC, USA

August 2023 - May 2027

TECHNICAL SKILLS

Programming Languages: C++, Rust, Python, x86-64 Assembly

Libraries & Frameworks: Boost, Cargo, FastAPI, CMake, SQLite, Raylib

Developer Tools: Git, GitHub, Linux/Unix, Visual Studio Code, Docker

Relevant Coursework: Operating Systems, Data Structures & Algorithms, Object-Oriented Programming, Design & Analysis of Algorithms, Database Management, Rust Programming, Agentic AI

PROJECTS

• **Castle Plus** | C++, Linux, Multithreading [GitHub](#)

- Built a from-scratch reverse proxy and process supervisor directly on Linux's epoll, terminating TLS and routing HTTP traffic to backend services.
- Added per-IP rate limiting, live certificate reloads without dropped connections, and a Unix-socket admin control plane for runtime configuration.

• **CPU Emulator** | Rust, C++, Python [GitHub](#)

- Designed a custom instruction set architecture from scratch, specifying 16 32-bit registers (dedicated frame, stack, and link pointers plus a program counter), variable-length 1/2/3/6-byte opcodes, 256-cell RAM, and a 65,536-byte disk.
- Wrote a two-pass C++ assembler that translates hand-authored .emu assembly into resolved binary output, computing label offsets for jump targets across a sample BIOS and test programs.
- Implemented a Rust emulator with a Raylib front end that executes assembled programs and renders live CPU, RAM, and I/O state through switchable normal, technical, and split debug views; paired it with a Python/Tkinter syntax-highlighting editor to complete a full assemble-execute-edit toolchain.

• **swiftSearch** | C++, Multithreading, File I/O [GitHub](#)

- Built a multi-threaded directory search tool that uses concurrent programming to scan large, user-specified directory trees efficiently.
- Implemented word extraction, numerical data sorting, and search operations, organized into classes for scalability and minimal performance degradation on large file systems.

• **bstd (B-Standard)** | C++, Rust, x86-64 ASM [GitHub](#)

- Built a ~7,000-line, dependency-free standard library spanning C++20 and Rust, organized under a namespaced (ds, encode, store, bit, form, syst) toolkit with zero third-party crates or packages.
- Implemented thread-safe containers (vector, queue, stack, hashmap, trie, B-tree) on shared_mutex reader/writer models, a from-scratch AES-256 cipher, and a Huffman coder with a custom bit-level buffer.
- Designed fixedDataStorage, a 2,700-line Rust append-only storage engine (3rd on-disk format generation) enabling lock-free concurrent writes via a single atomic row counter, with versioned headers and 5-bit character compression.

LEADERSHIP & ACTIVITIES

• **Association for Computing Machinery (ACM)** – Treasurer

August 2026 - Present

- Participate in and lead workshops, tech talks, and networking events to keep members current with industry trends
- Manage chapter budgets, coordinate event transit, and identify fundraising opportunities

• **ECU Coding Club** – Assistant Head Coach

August 2026 - Present

- Mentor students in competitive programming techniques and algorithmic problem-solving strategies
- Organize bi-weekly coding sessions and weekly Kattis practice problems

• **International Collegiate Programming Contest (ICPC)** – Team Member

September 2025 - Present

- Compete in regional programming competitions focused on algorithms, data structures, and efficient problem-solving
- Collaborate with teammates to solve complex computational problems under time constraints