Daniel Kirby

dankirby.dev@gmail.com | LinkedIn

Education

Queen Mary, University of London, BSc Economics *2023 - 2026*
Relevant modules: C++ for Finance (96%), Mathematical Methods (86%)

York College, A Levels *2021 - 2023*
Maths: A*, Economics: A, Computer Science: A

Skills

Programming Languages: C++ (20), C#, Rust, Python, Bash/Zsh Scripting, HQL
Systems & OS: Linux (Arch & Debian), Concurrency, Networking (Websockets)
Tools: Git, CMake, Perf (profiling), GDB (debugging), Google Benchmark, GTest (unit/parameterised testing)

Employment History

Technical Analyst Intern, Quantifi Solutions *July 2025 - September 2025*

- Optimised a C# data export utility handling tens of millions of records, cutting runtime by **over 90%** (29 to 2 minutes), significantly lowering resource load on the client's on-prem server
- Implemented concurrency across several internal and client facing scripts by **designing custom thread-safe data structures** to interop with the platform
- Diagnosed a system-wide performance bottleneck by debugging NHibernate calls, revealing a data-model flaw causing unnecessary query overhead
- Re-architected a Python script which automates a client's roll-forward process for their XVA calculation grid; achieving a **2x** speedup on a validation dataset

Freelance LLM Trainer, Outlier *February 2025 - July 2025*

- Evaluated LLM responses for accuracy, instruction following, and writing style, following project specifications.
- Generated high quality training data, specialising in Economics, Mathematics and Computer Science.
- Achieved an average quality score of 4.1/5 from over 30 reviews
- Promoted to **Senior Reviewer on 4 different projects** after demonstrating high quality work and adherence to guidelines

Projects

SIMD Monte Carlo Engine Repository *October 2025 - November 2025*

- Developed a high performance, concurrent Monte Carlo simulation engine in C++17, leveraging low level AVX/AVX2 CPU intrinsics to speed up path generation by **~30x** over traditional implementations
- Designed and implemented a custom, fully SIMD random number generator, using Xorshift128+, Box-Muller transform, and inverse normal CDF approximation to generate random numbers **~3.5x** faster than the C++ standard library
- Implemented a modular framework based on interfaces to allow easy addition of new exotic option types and underlying stochastic models
- Established a robust Linux development workflow, writing custom Bash build scripts and integrating CMake, Google Benchmark, `perf` for profiling, and `gdb` for debugging

Competitions

AlgoTrade Hackathon 2025 *June 2025*

- Invited as **1 of 100** international students to compete in the hackathon in Zagreb, Croatia
- Researched and implemented HFT strategies in Python and C++, focusing on market making and short straddle algorithms
- Engineered a low latency websocket client to stream and process live order book data, managing connection reliability and handling JSON data serialisation
- Developed standardised Python dataclasses for shared data models and implemented Git version control, enabling rapid collaborative innovation

IMC Prosperity *April 2025*

- Researched and developed strategies to trade fictional products, applying mean reversion, pair trading and market neutral strategies.
- Implemented trading algorithms in Python, introducing optimisations as more products were introduced over the 15-day challenge
- Placed in the **top 3% of over 12,000 teams as a solo participant**