Jasveer Dhillon

437.669.6942 | jasveerdhillon05@gmail.com | github.com/JasveerD | LinkedIn

EXPERIENCE

York University – NSERC USRA May 2025 – Aug. 2025
Undergraduate Researcher (Supervisor: Prof. Simone Pisana) Toronto, ON

- Conducted research on *"Using Graphene as a Material for Tritium and Deuterium Separation in Water"* under the NSERC Undergraduate Student Research Award
- Fabricated graphene oxide (GO) filters and prepared samples for structural and material characterization
- Trained to operate a Raman spectroscope; collected and analyzed spectral data to estimate degree of oxidation in GO membranes
- Processed raw spectroscopy data using Python and SciPy for curve fitting and peak analysis
- Presented research findings at a summer research conference

PROJECTS

Neural Network from Scratch | C++20, BLAS | GitHub

- Built a neural network library from scratch in C++20 with no ML frameworks; trained to **97.91% accuracy** on MNIST in 20 epochs
- Implemented a templated `Matrix<T>` class with heap-allocated row-major storage, full Rule of 5 compliance, and operator overloading
- Derived and implemented backpropagation analytically from first principles, with He initialization to eliminate vanishing gradients in ReLU networks
- Accelerated matrix multiplication via `cblas_sgemm` (Apple Accelerate/AMX), achieving up to **842× speedup** over naive triple-loop at 1024×1024
- Designed modular architecture separating layers, activations, loss, optimizer, and dataloader; added binary serialization for checkpoint save/resume

Boids Simulator | C++, Raylib | GitHub

- Built a real-time 2D flocking simulation in C++ using Raylib, implementing Craig Reynolds' boids algorithm
- Modelled emergent flocking behaviour from three local steering rules – separation, alignment, and cohesion – applied per-agent each frame with no global state
- Weighted and clamped steering forces per frame with configurable perception radii and force multipliers, producing naturalistic flock dynamics

Fishy – BASIC to C Compiler | Python | GitHub

- Implemented a source-to-source compiler from a BASIC dialect to ANSI C, comprising a hand-written lexer, recursive-descent parser, and code emitter
- Enforced correct operator precedence through a layered grammar (expression → term → unary → primary), without relying on any parsing library
- Tracked declared variables and labels in symbol sets; validated all GOTO targets post-parse to catch forward-reference errors
- Emitter maintains a separate header buffer to hoist variable declarations to the top of `main()`, conforming to ANSI C declaration rules

EDUCATION

York University Sept. 2023 – May 2028

B.Sc. Hons. Computer Science *CGPA*: 8.03/9

- **Coursework**: Data Structures, Algorithms, Databases, Computer Systems
- **Awards:**
  - *NSERC Undergraduate Student Research Award (USRA)* – Competitive national research award
  - *Lassonde Undergraduate Research Award (LURA)* – Competitive faculty-level research award
  - *Denise Hobbins Prize* – Highest average in first-year physics courses
  - *Dean's Honour Roll Member*

SKILLS

Languages: C/C++, Python, Java, Bash, x86_64/ARM/RISC-V Assembly, SQL (MySQL)

Tools: Neovim, Git, CMake, GCC/Clang, SLURM, JetBrains IDEs, Jupyter Notebooks, Linux

Libraries: NumPy, Pandas, Matplotlib, SciPy, Raylib, Catch2