

Michael Neuper

Stellenbosch, Western Cape

✉ michael@michaelneuper.com

☎ +27 82 062 1316

🌐 michaelneuper

🔗 michaelneuper

EDUCATION

Stellenbosch University | *B.Sc. Computer Science* **2024–Present**

Expected completion: December 2026. Achieved distinctions in all Computer Science and Mathematics modules.

Senior Merit Award, 2025: Ranked among the top 5% of senior students in the faculty.

Helpmekaar College | *National Senior Certificate; 9 distinctions* **2019–2023**

EXPERIENCE

Learning Assistant | *Stellenbosch University* **Jan 2026–Present**

Support students during practicals for the following modules:

- Introductory Computer Science (CS114)
- Computer Architecture (CS244)

Software Engineering Intern | *BBD Software* **Jul 2025; Jul 2026**

- Collaborated in four-person teams to design and develop two full-stack web applications during separate internship programmes.
- Developed a laser-tag game using computer vision for real-time player detection and tagging.
- Developed a foosball game controlled through smartphone gyroscope input.

Cybersecurity Consultant Intern | *MWR CyberSec* **Dec 2024**

- Completed cybersecurity training with hands-on labs on TryHackMe.
- Conducted a full web application security assessment, documenting vulnerabilities with risk impact and remediation guidance.
- Authored a professional report and delivered a client-style presentation to a panel of industry professionals.

PROJECTS

8-Bit Breadboard Computer | *74HC Logic, EEPROMs* **2026**

- Built a programmable 8-bit computer on breadboards from 74-series logic chips, following Ben Eater's design.
- Assembled modules sharing one 8-bit bus: a program counter, 16 bytes of RAM, an instruction register, A and B registers, an adder/subtractor, and a 7-segment output display.
- Programmed EEPROM microcode that drives sixteen control lines over a five-step instruction cycle, with carry and zero flags for conditional jumps.

Multicore OS Scheduler | *C, OpenMP* **2025**

- Built a multicore process scheduler in C and parallelised process execution using OpenMP.
- Implemented FCFS, Round Robin, and pre-emptive priority scheduling, together with resource management and process synchronisation.
- Added deadlock detection to identify unsafe resource-allocation states.

Compiler | *C, JVM* **2025**

Implemented a compiler in C for a custom programming language, including:

- A hand-written lexer and recursive-descent parser.
- A symbol table for semantic analysis/type checking.
- A code generator that emits JVM bytecode, enabling execution on the JVM.

Othello with Monte Carlo Tree Search | *C, MPI, Docker* **2025**

- Built an Othello player in C that selects moves using Monte Carlo tree search with UCT, under a one-second limit per move.
- Parallelised the search over MPI in three versions: independent trees combined with `MPI_Allreduce`, periodic statistic sharing, and asynchronous sharing with non-blocking sends.
- Packaged each version as a Docker image and wrote a match runner to benchmark the versions against each other.

TECHNICAL SKILLS

Languages: Python, Java, C, JavaScript, SQL, HTML/CSS

Technologies: React, Linux, Git, Vim, LaTeX