

# Dhruv Chawla

## Compiler Engineer

dhruv263.dc@gmail.com

+91 9910299843

<https://dc03.github.io>




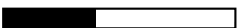





Vellore, Tamil Nadu

A capable programmer learning compiler design and the LLVM compiler infrastructure. Experienced with interpreters, and familiar with systems and assembly programming. 4 years of C++ experience.

### Languages

English - Fluent  
Hindi - Intermediate

### Skills

C++   
C   
Python   
Rust   
Compilers   
Interpreters   
Operating Systems   
Linux   
Git 

### Education

#### B.Tech in Information Technology

VIT, Vellore 2020 - 2024

Current CGPA: 9.28

#### XII (Senior Secondary), CBSE

Navy Children School, Mumbai 2020

Percentage: 96.4%

#### X (Secondary), CBSE

Navy Children School, Mumbai 2018

Percentage: 93.6%

### Experience

#### The LLVM Compiler Infrastructure

January 2023 - Present

*Examples of Patches Contributed*

- [clang][alias|ifunc]: Add a diagnostic for mangled names
- [NFC][TLI] Replace std::lower\_bound call in getLibFunc with DenseMap lookup
- [NFC][ValueTracking]: Remove redundant computeKnownBits call for LoadInst in isKnownNonZero
- [InstCombine] Fold icmps comparing uadd\_sat with a constant
- [SetVector] Improve performance for small sizes

#### Google Summer of Code Contributor

(The ENIGMA Team)

June 13, 2022 - September 12, 2022

*Project name: Data Buffers / Serialization*

- Worked on rewriting most of the frontend of the ENIGMA Development Language compiler, a scripting language based on GML
- Rewrote most of the Binary Buffer system which deals with storing and reading data from byte streams
- Made a serialization and deserialization system which uses template metaprogramming for static polymorphism
- <https://summerofcode.withgoogle.com/programs/2022/projects/BrXiUNA2>

---

## Projects

<b>nyx</b> <ul style="list-style-type: none"><li>- A simple, interpreted language implemented in C++</li><li>- Features classes with constructors and destructors, lists, tuples</li><li>- Static type system</li><li>- Copy, reference and move semantics</li><li>- Bytecode virtual machine</li><li>- Code formatter, bytecode dumper, VM execution tracing</li></ul>	<a href="https://github.com/dc03/nyx">https://github.com/dc03/nyx</a>	<i>September 2020</i>
<b>rispy</b> <ul style="list-style-type: none"><li>- Interpreter for a lispy-inspired lisp</li><li>- Implemented in Rust</li><li>- Tree-walk interpreter</li><li>- Testing for lexical analyzer and parser</li></ul>	<a href="https://github.com/dc03/rispy">https://github.com/dc03/rispy</a>	<i>February 2022</i>
<b>tictactoe-arduino</b> <ul style="list-style-type: none"><li>- Tic-tac-toe implemented on an Arduino Uno</li><li>- Multiplexing of outputs (LEDs) and inputs (buttons) to reduce pin usage</li><li>- Compact layout of game state to reduce memory usage</li><li>- Part of a university course project</li></ul>	<a href="https://github.com/dc03/tictactoe-arduino">https://github.com/dc03/tictactoe-arduino</a>	<i>February 2023</i>

---

## Certifications

<b>Introduction to Haskell Programming (NPTEL)</b> Percentage: 85%	<i>Issued Sep 2022</i> <a href="#">Credential ID NPTEL22CS69S2318078809012045</a>
<b>Compiler Design (NPTEL)</b> Percentage: 90%	<i>Issued Apr 2022</i> <a href="#">Credential ID NPTEL22CS14S2446142802071248</a>
<b>Design and Analysis of Algorithms (NPTEL)</b> Percentage: 85%	<i>Issued Oct 2021</i> <a href="#">Credential ID NPTEL21CS68S4332059403122958</a>