ARTIN GHASIVAND

Homepage: artingh.com
E-Mail: ghasivand.artin@gmail.com
Haskell Gitlab: gitlab.haskell.org/Ei30metry ◇ **Github**: github.com/Ei30metry

EDUCATION

Alborz High School *January 2023*
Diploma in Mathematics & Physics
Member of Biology Olympiad class

EXPERIENCE

Cryptal Global June 2022 - December 2022
Junior Developer *Dubai, UAE*

- Automation bots in Python
- DevOps and Linux Administration

VOLUNTEERING

Haskell Foundation Feb 2023 - Present
Podcast quality inspector

- Work with the editor and the hosts of the Haskell interlude podcast to make sure the episodes don't contain quality problems like noises or repetitive dialogues.

PROJECTS

Implementation of the GHC formalism paper (WIP): Typechecker for Haskell
A fresh Haskell-typechecker built on top of the formalism described in the Typing of GHC Haskell, Part I (Early Draft). Artin Ghasivand, Simon Peyton Jones and Richard A. Eisenberg paper, with key focuses of being: A one-to-one match to the inference rules, a robust way to test the underlying formalism, and a pedagogical tool for teaching about type-checkers and implementing them.

Blueprint (WIP): Pretty-print outgoing call-hierarchies of Haskell functions
The project uses the Glasgow Haskell Comiler as a library to query the internal abstract syntax tree representation of Haskell functions and pretty-print it to the user. Although the project isn't finished yet, it helped me get familiar with GHC's internals, got accepted as one of Zurihac 2023's projects , and had a key role in helping me attend GHC 2023 Contributors' workshop.

Hygeia (WIP): A CLI program to keep track of your moods
The project was initially started with the sole purpose of writing an actual program in Haskell that parsed a text file containing a special format with feeling entries, do some summarization, and report the pretty-printed output to the user.

Key educational points of the project: A text parser written using parsec, a command line argument parser using optpars-applicative, a pretty-printer using prettyprinter, and managed effects using the monad transformers library. The older versions also used the singletons library, which was later removed to simplify the codebase.

TALKS

A good programming language is a *Functional* one, University of Tehran. (slides, video)

TECHNICAL STRENGTHS

Programming Languages	Haskell, Agda, Racket, Emacs Lisp, Common Lisp, Python, C
Other Languages	LaTeX, ott
Tools	Git, Nix, GNU Make, Docker