

# Maxim Mints

+1 (202) 830-9729 – [97mints@gmail.com](mailto:97mints@gmail.com) – [mints97.github.io](https://mints97.github.io)

## OBJECTIVE

I am interested in designing low-level computer system software, building powerful foundations for wide arrays of applications.

## EDUCATION

### Georgia Institute of Technology

Atlanta, GA

- **Bachelor of Science in Computer Science** (Threads: Systems & Architecture / Information Internetworks)
- **GPA: 4.0; Started: Aug 2015; Expected Graduation: Dec 2018**
- **Relevant Courses Taken:** Operating Systems, Compilers & Interpreters, Processor Design, Computer Networking

## WORK EXPERIENCE

### Amazon ([amazon.com](https://amazon.com))

Seattle, WA

**Job title:** Software Development Engineer Intern

**Period:** May 2018 – Aug 2018

- Assisted with integrating Alexa Smart Home with security panel systems by improving Alexa's voice control capabilities.
  - Enhanced voice commands by creating an extensible plugin-based architecture where one can define strategies of different priority that attempt to algorithmically match the processed input data to some specific voice control target type.
  - Used the plugin-based architecture to add support for targeting security panels by brand name with voice commands.
  - Designed and proposed a more powerful long-term solution for targeting by brand name, directly using the Alexa natural language processing system for analyzing input, but sacrificing some flexibility.

### Hughes Network Systems ([hughes.com](https://hughes.com))

Germantown, MD

**Job title:** Software Engineering Intern

**Period:** May 2017 – Aug 2017

- Assisted with the development of a novel algorithm for classifying network flows using throughput-based metrics.
  - Wrote an efficient tool in C++ that converted each flow in several packet capture files into a throughput time series.
  - Fully implemented the network flow classification algorithm in Python, and tested it on throughput time series from videos of different resolutions, reaching 73.3% classification accuracy.
- Created a complex tool to detect potential sources of interference in the signals received by satellite dish terminals.
  - Used Python to efficiently automate multi-threaded collection of signal-to-noise values from over 52000 terminals.
  - Developed an algorithm for approximating potential locations of interference sources and implemented it with Java.

### Georgia Institute of Technology ([gatech.edu](https://gatech.edu))

Atlanta, GA

**Job title:** Undergraduate Research Assistant – Trustable Programming Group

**Period:** Aug 2017 – Ongoing

- Implemented, in Haskell, an efficient and configurable rewrite-rule simplifier for quantifier-free linear integer arithmetic formulas ([github.com/Mints97/rewrite-simplifier](https://github.com/Mints97/rewrite-simplifier)), to be used as an optimization strategy in a model-checking algorithm.
- The model-checking algorithm, created by Professor William R. Harris and his Trustable Programming Group, uses relational invariants between program states to prove or disprove partial functional equivalence ([gt-pequod.github.io](https://gt-pequod.github.io)).
- On certain benchmarks, the formula simplification optimization produced a 200% speed gain.

**Job title:** Undergraduate Teaching Assistant – Design & Analysis of Operating Systems

**Period:** Aug 2018 – Ongoing

- Creating reference implementations for the project assignments; grading, helping the students understand the material.

## SKILLS

- **Programming Languages:** Haskell, C, C#, Java, C++, Python, SWI-Prolog, Assembly (x86: GAS, FASM)
- **Hardware:** Verilog, VHDL, Altera Cyclone V FPGA, Arduino
- **Spoken Languages:** Russian – native, English – fluent, Spanish – basic

## PERSONAL PROJECTS

**Alcolang** (May 2015 - Ongoing) [github.com/Mints97/alcolang](https://github.com/Mints97/alcolang)

- An in-development, interpreted, pure, reactive, prototype-oriented, strongly and dynamically-typed, lazy, pattern-matching, multi-paradigm programming language. A implementation in Haskell is currently a work-in-progress.
- A powerful programming platform over extremely simplistic abstractions and the smallest number of lexical elements.
- This is achieved by parsing expressions lazily and performing a text equivalence check during pattern matching: so, an unparsed element can be a comment, string, numeric literal, keyword, or a name, erasing boundaries between these concepts.

**Y Window System** (Apr 2018) [github.com/Mints97/yws](https://github.com/Mints97/yws)

- Tile-based window system for the xv6 operating system with mouse-controlled resizing and image display capabilities.

**CControlFlow** (Feb 2015 - Jul 2015) [github.com/Mints97/CControlFlow](https://github.com/Mints97/CControlFlow)

- C# library that helps find errors in code by generating control-flow graphs of programs written in C (C89).

**tinyObject** (Mar 2014 – Dec 2014) [github.com/Mints97/tinyObject](https://github.com/Mints97/tinyObject)

- C framework which enables writing object-oriented code with true inheritance in lieu of the regular composition approach.