Curriculum Vitae

James Hobson

August 2, 2024

Summary

I'm currently one year into my masters in informatics, and working part time in the R&D department at Jotron AS. In my short time at work, I have already applied myself in a large number of fields, ranging from Linux Kernel development and bootloader programming, compiler development to computer vision and websites. My heart lies in formal methods and correctness and this is what I think defines my *style*. Applying mathematical logic to my code is what has allowed me to quickly make projects that stretch from the kernel to a pretty web server.

Skill Highlight

- Proficiency in C/C++, Haskell, Scala, C#, F#, Scheme, OCAML, BASH, Assembler (ARM and X86_64)
- Ability to quickly understand large, complex systems
- Proficiency in Linux and other UNIX Systems
- Knowledge of LaTeX and experience using it for technical and academic writing
- Proficiency in applying formal methods

Education

Uppingham School: 2012-2015. (UK) Abbey Grange Academy: 2015-2017. Oxford University: 2017-2020. University of Bergen: 2023-2025

- University of Bergen (See appendix for modules and grades)
- Oxford University, Computer Science and Philosophy: 2:1 (See appendix for modules and grades)
- **A-Levels:** Maths: A*, Physics: A, Economics: A, Further Maths: B
- **AS-Levels:** Further Maths: A, Physics: A, Economics: A, General Studies: A, Maths: B
- **(I)GCSEs:** 9 A*s, 1 As and 2 Bs – A*s include Maths and Physics. Also Further maths additional qualification level 3: A

Experience

Jotron (Norway) 12-Week Contract – Summer 2020 - Present

- I research and prototype new software products and features for use in air traffic control
- Architect of our linux distribution project and driver author
- Main developer of the Jotron SW Screen recorder
- Devops for linux software delivery
- Computer Vision projects for analysis of air traffic control footage

Teaching at University of Bergen

- TA position for the advanced function programming course. Gave session and occasional lectures on Monad Transformers, parsers, algebraic effects, graphics
- TA manager for introduction to functional programming. In charge of writing exercises and assessing students. As well as running interventions to ensure that no student is left behind
- TA for introduction to functional programming. Running weekly sessions, helping students with their exercises

Jotron (Norway) 12-Week Contract – Summer 2019

- I was set the task to build a screen recorder fit for air traffic control, an environment where safety, data integrity and performance were paramount
- I was the only developer on the project, having control over the entire code base.
- The solution used abstract algebra to model data pipelines that could be rearranged and simplified to achieve maximum performance whilst guaranteeing that optimisations wouldn't result in unintended behaviour
- Jotron have since hired me as a consultant for R&D

ARM (UK) 12-Week Internship – Summer 2018

- Built a project in C (and some Haskell) to do data capture and analysis as part of the Mali Display test framework
- Independent project with a set end goal but no information past the specification and only me as a developer
- Experienced working on a semi-research project, where the final product matters as much as the proof of concept
- Built tools for specific data analysis (which I have been told were actually useful!)
- Experienced getting up to speed with technology I had not even heard of before

Backend Web Development Summer Job at Enjoy Digital (UK) – Summer 2017

- Building CMSs that could be used by clients to customise and create pages with no risk of breaking their webpage
- Elastic Search integration for websites – extracting key data from multiple sources and arranging it in complex data types that can be queried in flexible ways (No-SQL)
- Independent research project into lightweight machine learning for film recommendations
- Independent research project into population modeling to optimise website cache
- Wrote several tools still used by the business today

Game Development Uppingham Summer School (UK) – one week each summer 2013-2015

- Teaching assistant on a week-long summer school for people aged 11-18 on game development using Unity, Blender and other software
- Worked in C#, building teaching materials that students could use in their games: generic scripts that they could easily modify to learn the thought behind algorithm design without needing much C# knowledge
- Experience in communicating technical ideas clearly, as many of the students had only little technical experience
- Experience working with people with various disabilities
- Experience with tight deadlines (one week to build many games)

Personal Projects

Ever since I started to code, I've been building small projects. Here are some highlights:

- pi-radio, an embedded audio player with a fair web queuing system (Haskell, QML)
- qml-hs, graphics framework (Haskell)
- Open Source Parser combinator library (C++)
- Documentation and packages for the openSUSE Linux project
- Contributions to the linux appimage project (c)
- Youtube tutorial series on the haskell programming language
- CTL Model Checker (Haskell)
- A Distributed Computing Platform where phones solve small parts of large problems when charging at night (Won Oxford University Prize)
- Compiler written in Haskell to compile contracts with a web form as input
- DIY Smart Lock for room in college (c)
- Set cryptographic challenges in the College Newspaper (mostly Haskell, BASH, JS, HTML)
- Made C programming tutorials for RISC OS (to be found on Youtube)
- Designed 3 simple microprocessors in Logisim (drag and drop logic gates to build reusable components)

Appendix

University of Bergen Subjects and results (A-F or Pass/Fail)

- Advanced Functional Programming: A
- Category Theory: A
- Introduction to Logic: A
- Software Security: A
- Elements of Programming Languages: A
- Selected Topics in Programming Theory: Pass
- Dependent Type Theory: Pass

Undergraduate at Oxford subjects

Subject	1st Year	2nd Year	3rd Year
Computer Science	Functional Programming, Imperative and OOP Programming, Design and Analysis of Algorithms, Discrete Maths, Probability	Compilers, Models of Computation, Lambda Calculus, Algorithms 2, Concurrent Programming Computer Security	Formal Verification, Principles of Programming Languages, Quantum Information, Computer Architecture 2
Philosophy	Into to Logic, Elements of Deductive Logic, General Philosophy Analytic and Historical, Turing on Computability & Intelligence	Philosophic Logic, Philosophy of Logic and Language	Thesis on Logical Consequence and Completeness of Formalist Systems