Danielle Mustillo

(514) 570-1479 https://danielle-mustillo.github.io
danielle.mustillo@mail.mcgill.ca

EDUCATION

Bachelor of Software Engineering – McGill University 2012-2016

- GPA: 3.73
- $500 scholarship – The Dean's Honour List (top 10% of Faculty)
- 99% final grade - Introduction to Computing (Java)
- Offered $1000 scholarship to identical program at Concordia University.

WORK EXPERIENCE

Coop Intern – Ericsson Canada – 4 Months 2016

- Developed Test Automation and Continuous Integration tools
- Python with Jinja templating, YAML files
- Setup reverse proxy w/ Apache Server to get around CORS for HTTPS REST queries
- Frontend JavaScript (JQuery, Vue.js), CSS, HTML: demoed these for users and designed to be interactive (debouncing, asynchronous, etc).
- Java, Intellij, Jenkins, Sonar (code coverage), Git, Gerrit Code Review, Vagrant, Unix, CentOS, Red Hat Linux, Ansible.

Coop Intern – Ericsson Canada – 4 Months 2015

- Developed IoT (Internet of Things) framework
- Java, Restful services, Gstreamer, Raspberry Pi
- Developed JavaScript/AngularJS/JQuery(testing) WebRTC webpage for P2P
- Created CoAP-like protocol for communication
- Interconnected IMS core with application (WCG) via REST, SRTP, SCTP/DTLS, UDP, TCP
- Developed target deployments through Bash & Python
- Used SCRUM, JUnit, Eclipse, Maven, Jenkins, Sonar, Ubuntu

Engineering Intern - Mannarino Systems & Software Inc. – 8 Months 2014

- DO-178-B trained
- Verification of flight production SW (in C).
- Test environment in Python, iSystem debugger for breakpoints.
- Modified Coverage/Decision Coverage and End-to-End testing of SW.

IT Student – Canadian Revenue Agency – 3 Months 2013

- Used Powershell scripts to automate remote admin tasks
- Fixed varying hardware issues (printers, scanners, software, etc).

TECHNICAL PROJECTS

- **Personal Website**
  - Jekyll (Ruby) with HTTPS for static webpages
  - Creative Commons
- **Machine Learning in LabView**
  - Developed a neural network in Labview to learn accelerometer positioning
  - Customized the net with bias, different training amounts, different number of layers, etc.
  - Unit tested these nets
- **Investment Portfolio Tracker**
  - Python, JavaScript, CSS, HTML with Libraries
- **Research**
  - SQL, MatplotLib & Python to interpret relations between data sets
  - Some Excel and formulas
- **Password Cracker**
  - OpenCL run on AMD R9 290 card
  - Multi-platform run on CPU and GPU to compare (10-20x speedup)
  - I had adapted the MD4 hashing algorithm for OpenCL (you can't use malloc for example)
- **Packet Sniffer**
  - Processes raw byte-code packets in C and extracts IPv4 headers and content data.
  - Attempted to run in on RJ-45 connector soldered to board, but chip not fast enough for real time processing (darn)
- **File System**
  - Functional filesystem with implemented inodes
  - Used Linux FUSE (Filesystem in Userspace) to expose APIs
  - Relatively complete support of Vim & Nano in this FS
- **Vector Calculator in Android**
- **Arduino Wall Follower**
  - Implemented low pass, median, out of bounds rejection filter to process real time data
  - Used ultrasonic sensors & motors to do this (low cost)
  - Developed connection algorithm to Python computer program to output CSV data for processing
- **Java Light Racer**
  - Build process using Ant
  - Used I/O streams, JUnit and created UML and Use Case Diagrams
  - Black box testing, white box testing, SOLID principles.
- **NXT Lejos Robot**
  - Developed search, pickup, drop off and obstacle avoidance algorithm
  - Concurrent Programming with threads (synchronization, dependency resolving, etc).
  - Data handling (P2P I/O streams between two different Lejos)
  - Event driven programming (listeners, timers, etc), Exception handling
  - Robotic concepts: Navigation, obstacle avoidance, odometry, odometry correction, search and rescue, localization.