

# OWEN DURKIN

Quincy, MA · [owen.c.durkin@gmail.com](mailto:owen.c.durkin@gmail.com) · 7742082533 · <https://github.com/ODurcain> |  
<https://www.linkedin.com/in/owen-durkin/>

## EDUCATION

### University of Massachusetts Boston

BS Computer Science, Minor in Mathematics *GPA: 3.4*

Boston, MA

Sep 2022 - Dec 2024

### Benjamin Franklin Institute of Technology

AS Mechanical Engineering Technology *GPA: 3.86*

Boston, MA

Sep 2016 - May 2018

## SKILLS

Programming Languages: C, C++, Java, Python, x86 Assembly, R, JQL, Racket  
Tools: Linux, Arduino, IntelliJ, Altium, Jira, Solidworks, VScode, PuTTY, WSL  
Ubuntu, RedMine, SQL/JQL, DrJava, DrRacket, GitHub, GitLab

## PROJECTS

### Portfolio Website *HTML, CSS, VScode, Github Pages* <https://odurcain.github.io/>

Created a web page based portfolio to introduce myself, my skills, and my projects as a glance.

### Huffman Encoding *C, VScode, Powershell*

Encoded files using Huffman tree logic. Led to a 40% decrease in file size.

### Bash Shell Terminal *C, VScode, Powershell*

Created a shell that allowed the use of env, setenv, unsetenv, cd, history, and pwd. Is able to parse user inputs, perform I/O redirection, and has a pipe.

### Queueing System *C, VScode, Powershell, PThreading library*

Used PThread library to implement multithreading for a queueing system that also allowed multiple servers and returned relevant time data.

### UNO Card Game *Java, IntelliJ*

Created a simplified UNO card game that had 4 AI players go against each other

### Dominoes *Java, IntelliJ, HTML, CSS*

Created a 2-4 player dominoes game with all traditional rules and functionality of dominoes in place.

### AI Model - NATOPS hand signal classification *Python, KMeans, Matplotlib, SKLearn, Pandas*

Created a learning model that clustered data based on KMeans and provided ROC curves through Gaussian Naive Bayes, Logistic Regression, K-Nearest Neighbor, and Random Forest classifications

### Hamming Encoding and Decoding *C, VScode, Powershell*

Used Hamming logic to encode and decode files. Process involved taking 4 characters at a time, converting them to ASCII giving 8 digits. Each digit was converted to binary and parity bits were derived from there. Used the new P1, etc digits to form a new ASCII value which was the compression. The decoding did this in reverse.

## EXPERIENCE

### Vicarious Surgical

*Senior Integration Engineering Technician*

Waltham, MA

Oct 2021 - Sep 2022

- Solidworks, Motion Capture (OptiTrack), control systems, Altium, Jira
- Python shell scripts, verification & validation, testing, troubleshooting, prototyping, cadaver labs, clinical trials
- Interdepartmental work aiding in combining the work of mechanical, software, and electrical engineering teams into a final product

### Vicarious Surgical

*Medical Robotics Technician to Technician Lead*

Waltham, MA

Aug 2020 - Oct 2021

- Engineering prototyping of vision system, robotic instrument building, troubleshooting, and testing
- SMD soldering 0201 components, through-hole soldering, hot air and soldering iron, and microscopic welding/soldering/assembly
- Management of technicians, investor demos, cadaver labs, and engineering prototyping accomplishing consistent meeting of deadlines
- Improvements on error analysis, and initial assembly leading to an overall increase in single use lifespan of 30%