

# McKenzie Bourn

bournm@oregonstate.edu • <https://github.com/kenziebourn> • <https://www.linkedin.com/in/mckenzie-bourn/>

## EDUCATION

---

**Oregon State University | Corvallis, OR**

**Expected Graduation: May 2024**

*B.S. Computer Science*

*Cumulative GPA: 3.72*

**Coursework:** Data Structures & Algorithms, Discrete Mathematics, Web Development, Differential & Integral Calculus, Computer Architecture & Assembly, UI/UX Engineering, Machine Learning, Engineering Computation and Algorithmic Thinking, Engineering Design and Problem Solving

## SKILLS & TECHNICAL TOOLS

---

Languages: Python, JavaScript

Technologies: Git, React, Node.js, PostgreSQL

## EXPERIENCE

---

**Medical Device Assembler | Terumo Blood & Cell Technologies**

June 2021- August 2021

- Worked alongside the Engineering department and memorized over twenty various Medical Operation Procedures (MOPs) in order to build different apheresis kits (TRIMA) for clients while also paying attention to successive checks on each product's subassemblies
- Performed area clearance while verifying Medical Equipment Documentation (MEDs) and complying with regulatory government guidelines i.e. OSHA, ISO, GMPS, FDA, etc. Applied use of pressure gauges, calipers, flow meters, micrometers and indicator dials as well as implementing ergonomics into daily routine

## PROJECTS

---

**E-Commerce Website | React JS**

- Developed and designed a mock e-commerce website for my personal clothing brand from scratch using React JS
- Utilized React components and hooks, as well as the Tailwind CSS's HeroIcons library to manage state and build out functionalities such as product listings, reviews, email sign up and product search. Implemented these features to simulate a real online store experience.
- Deployed the application using Cloudflare to ensure secure and efficient hosting
- This website is based on my actual Depop (online marketplace platform) seller profile and has data extracted from actual products that have been sold as well as real-time reviews.

**Mancala Game | Python**

- Built the classic Mancala board game that allows two people to play a text based version of the game
- Implemented utility methods for traversing the board, getting and setting the number of seeds in a pit or store, and checking the validity of a pit index

- **Object-Oriented Programming:** The program is organized into classes, each representing a specific aspect of the Mancala game (e.g. the game as played, a player, the board). These classes contain attributes and methods that represent the state and behavior of the Mancala game. It correctly simulates the rules of the game, including handling player turns, sowing the seeds and determining the winner
- **Data manipulation:** The program involves manipulating data structures such as lists and dictionaries to store and retrieve information about the state of the game, including the number of seeds in each pit and store and the names of the players.
- **Debugging and error handling:** includes custom exception classes and error checking to handle invalid input and other potential errors that could occur during gameplay

### **Valet Simulator | Python**

- **Object-oriented programming (OOP):** The program is organized into classes, each representing a specific aspect of the valet service (e.g. a vehicle, a car, a motorcycle, the garage). These classes contain attributes and methods that represent the state and behavior of the valet service.
- **Design and implementation accurately simulates the valet service,** including adding and removing vehicles from the garage, displaying the vehicles in the garage, charging the appropriate rate when a vehicle leaves, and allowing overnight customers to request their car
- **Data manipulation:** The program involves manipulating data structures such as lists and dictionaries to store and retrieve information about the state of the garage, including the number of spots available, the vehicles currently in the garage, and the vehicle information for each vehicle.
- **Utilizes custom exception classes and error checking to handle a full garage and other potential errors that could occur during the valet service**