From classes I took while at OSU. These were all listed as portfolio projects when I took them and are able to be listed here. The class name is listed in the outer bullet, the folder name is the second bullet.

- CS 162 - Introduction to Computer Science
    - Hasami Shogi
        - Allows the user to play the game Hasami Shogi against another player on the same pc using Python. There is an example of testing out the game commented out at the bottom (this was before i knew about checking if main was run)
        - https://github.com/osu-cs162-f21/portfolio-project-hamrangj
- CS 271 - Computer Architecture and Assembly Language
    - Assembly Project
        - String primitives and Macros in assembly
            - Tasked with writing an MASM program that uses string primitives and macros in assembly to get 10 integers from the user and output the average and other info, while dealing with invalid input
            - https://canvas.oregonstate.edu/courses/1876763/assignments/8575288/submissions/6472405
- CS 290 - Web Development
    - Web Development
        - A full Stack Mern App using mongo db, javascript. Organized into a React app folder and a REST API
        - Uses some boilerplate code from here: https://canvas.oregonstate.edu/courses/1849583/assignments/8703880?module_item_id=21730853
- CS 261 - Data structures
    - HashMap
        - Implemented in Python in open addressing with quadratic probing (hash_map_oa.py) and separate chaining (hash_map_sc.py). Run either depending on what you want to see
            - Uses some skeleton code from here (mainly the a6_include.py): https://canvas.oregonstate.edu/courses/1849559/assignments/8706727?module_item_id=21738475
- CS 340 - Introduction to Databases
    - Databases
        - A database using MYSQL that can be modified and shown on webpages. Created with my partner for the project, John Cheng. I worked on the Players, Games, and Games_has_Teams entities, along with about half of the Python code. This project uses Flask, but I can't host it anywhere as I no longer have access to certain tools that were used to make/host it, but the code is uploaded here.
- CS 372 - Introduction to computer networks
    - Networks
        - A client-server chat using sockets in Python

            - Run server.py first, then client.py. To exit type /q in each window
- CS 361 - Software Engineering 1
    - Software Engineering and Microservices
        - Made a simple text game that works with a microservice in Python. The main code was made by me, but a microservice in the game to handle fights (combatserver.py) was made by John Cheng
        - To run put both in a folder and run Game_milestone_3.py
            - https://canvas.oregonstate.edu/courses/1879201/assignments/8918529/submissions/6472405
- Cs 467
    - Machine Learning Breakout
        - My classmates,Aaron Bertell, Vince Wyborski, and myself (hamrangj in the linked repo) were tasked with making a version of the Atari game breakout with a machine learning counterpart made with Unity and the Unity MLAgents addon. I worked on the menus, scene management, win/loss condition setup, lives trackers, WebGL build, and the original version of the ML vs Player level design. You can play it on the second link. (A moves the paddle left, D moves it right)
        - https://github.com/GitAarontell/MLBreakout
        - https://gitaarontell.github.io/MLBreakout/