

Instructions:

The program's main.py should be run from inside the Health-program-project folder. The program's structure is pictured below. The program functions as a basic health assistant. You can ask it to generate a workout based on your specifications, you add exercises to different workouts, log meals, and get a print out of all your health activities organized by date. If you want to create a workout select option 1 from the main menu, to add a new exercise to a workout group, select option 2. If you'd like to log a meal, select option 3. To view all of your health activity select option 4. The program will also check for invalid entries and prompt the user to enter correctly.

- Program Structure:

```
health-program-project/  
├── main.py  
├── meal  
│   ├── __pycache__  
│   │   └── meal.cpython-36.pyc  
│   └── meal.py  
├── workouts  
│   ├── __pycache__  
│   │   ├── core.cpython-36.pyc  
│   │   ├── hiit.cpython-36.pyc  
│   │   ├── strength.cpython-36.pyc  
│   │   └── workout.cpython-36.pyc  
│   ├── core.py  
│   ├── hiit.py  
│   ├── strength.py  
│   └── workout.py
```

Changes and improvements:

Originally, I planned on creating a Yoga class but realized that a yoga workout progresses in a way that's different from other workouts because it isn't just a certain number of reps or timed. It was difficult to generate a yoga workout in a way that it would actually flow. If given more time, I would work through this issue and add the Yoga class with and allow users to select different types of flows. I would also add the capability for users to log exercises outside of the preset child classes, i.e running, swimming, etc.

Challenges:

It was a little difficult determining the best way to combine the functionality of the classes in the main.py with the user inputs. I wanted the user to have a lot of options, but realized every option required me to build out an entirely new functionality, so in the interest of time I ended up only including options/functions that were integral to the program. It also took me a while to optimize usability and ensure the printout statements were clear and organized nicely. I also hit a snag implementing some of the error handling, I was finding that it would sometimes disrupt the functionality of the program.