# Create a short video (that’s no more than 3 minutes in length) in which you present your project to the world. Your video must begin with an opening section that displays:

#

# your project’s title;

Mike’s Workout Tracker

# your name;

Michael Swithers

# your GitHub and edX usernames;

GitHub – strife271

Edx username - Mike8957

your city and country;

Raleigh, NC, United States

# and, the date you have recorded this video.

8/9/2024

# It should then go on to demonstrate your project in action, as with slides, screenshots, voiceover, and/or live action. See howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen for tips on how to make a “screencast,” though you’re welcome to use an actual camera. Upload your video to YouTube (or, if blocked in your country, a similar site) and take note of its URL; it’s fine to flag it as “unlisted,” but don’t flag it as “private.”

# Mike's Work Out Tracker  
#### Video Demo: <URL HERE>  
#### Description:

#### Overview:

This project gathers data from users about their workout routines. The user could also chose to load workout data from a comma separated value file. Then it can display the data to the terminal and plot a chart that shows a user’s exercise progress over time. Also, it includes an option to update the data and save it as a file.

#### Files:

project.py - includes all the python code to run the program

test\_project.py – pytest functions to ensure the functions are working correctly

sample\_file.csv – needed to run the pytest functions

README.md – description of project - this file!

requirements.txt – pip installed python packages required

#### Use:

After running project.py a command line interface opens. There are seven options in the cli:

1. Start new workout routine

The user can input a new routine. Including: name, routine name, exercises with exercise name, reps, weight and sets.

1. Load workout from file

Get previously saved data from a .csv file.

1. Display workout routine on screen')

Displays the workout data to the terminal in tabular form.

1. Add to workout routine

Updates more exercises to the routine

1. Plot weight over time

Displays a chart with a user’s progress over time

1. Save workout routine to file

Saves the data to a .csv file.

1. Exit

Exit the program.

I created two custom classes to make this program. The Routine class which has a list of exercises, and an Exercise class which contains the data for each exercise completed. The Exercise class has the name of the user, the type of routine (such as chest, legs, or back), the date the routine (generated by the system), the exercise name, the number of repetitions, the amount of weight lifted, the number of sets, and the total weight lifted. I also created several functions to implement the program, so I used a combination of functional and object-oriented programming in order to get a better understanding of both.

#### Testing

To test the program, I used the Python package pytest. I created three test functions. The first test that the total weight is calculated correctly. The second test determines if the data to plot the chart is retrieved correctly. And the third test is to determine if the data is read correctly from the comma separated file.

#### Conclusion

This was a challenge for me as I am new to programming. In hindsight I would have simplified the program. I would also add have added a graphical user interface with dropdown menus. In future versions I would refactor the script to be easier for the user and include additional analysis and functionality.