Tabitha Geiger

CS 162

Prof. J Jess

Project 2

1. How do you think you would add file reading and writing functionality to your menu program?

Like in your videos, you can take the information from certain attributes and write them into a text file. For your example, you wrote a Person class with name, age, and if they were alive or not. For mine, it’s a House class with the main attributes being square footage of each room: livingroom, kitchen, bedroom, bathroom, respectively.

Writing these into a file would include the attributes (the square footage) of each room into that text file. For each room, I had 100 sq. feet (I should actually have changed the variation a bit just in case those values weren’t correct or it was assigning one variable’s value to the others.) That information would be written into the text file and that will then be stored for use when I need it.

Reading lines could be useful when we want to go through the specific line(s) that we need or to append additional square footage that is included at a later time.

We use try and except to catch errors. Try is used to run the function if there is no exception. If an exception occurs it skips the try clause and raises the error. These clauses are used to test for certain errors. The file-not-found error is raised when the file or folder you’re trying to open or write to does not exist.

1. I wanted to make some functions within my code that could be used to determine how many bedrooms and possibly bathrooms using user input. By doing so, that data could be added to a list variable where they can be accessed. This data could also be used to write into a new text file that can be accessed and read/stored later.

I also realized that my sq foot module could be broken up into two. One to make the formula and return the value, the other to print those values. For this week, I’ll be remaking that in order to test my code.

1. I attempted to read and write through Leland Wendel’s Motor class and was successful. I read through the self.aspiration attribute and also went through his get\_horsepower() module to read and write that value to a text file.

I believe a user input function could be useful as it is mentioned in their code that it is required. This user input could be used to add value to the attributes to the \_\_init\_\_ class. A print function would be useful in order to show that data to the user if they would like to see the values of horsepower, redline, and cost in total.

We could verify values through testing the values returned by the get\_horsepower(), get\_cost(), and get\_redline() functions by inserting the expected value and asserting the actual module to determine whether these modules are working properly. I could also try and insert parameters that aren’t the expected values like a string for an int parameter or an int for a string parameter when creating the class because the self.aspiration requires three specific strings in order for the rest of the modules to work.

As for PEP standards, I think it’s written well compared to style conventions. However, in the attributes of the \_\_init\_\_ section of the class, one of the parameters that is required is “code” and I don’t know what that’s referring to or where it’s used in the main code. A comment explaining why this specific parameter is needed would be beneficial for anyone looking at this code.