Leland Wendel

CS162 - Project 2

1. How do you think you would add file reading and writing functionality to your first week's project in this class?

    I added file reading and writing functionality to my first week's project to provide the customer with an invoice for their engine build. The program can also print old invoices if given the invoice number.

2. How do you think the **try** and **except** clauses of Python work (based on your current understanding or a quick Internet search) and what the **FileNotFoundError** class is.\

    Try and except clauses are used to handle errors at runtime that would otherwise cause the program to stop running. The FileNotFoundError occurs when the program cannot locate a required file, usually due to an incorrect file path or user input. This error could be handled with a try and except clause – the program could ask the user to retype the filename or specify the complete file path, instead of crashing.

3. Identify sections of your code that you could use to create functions or methods out of.

    I'm happy with the methods I've used to define the motor so far. I think my new_motor() function could probably be simplified by having another function handle the file writing.

4. Design tests that would show your program working correctly (positive tests and negative tests).

    Incorrect integer input and FileNotFound errors are handled correctly. I've learned in testing that letter and float point inputs are not handled in the main menu function, and this will halt the program.

5. **Attempt** to create a couple test functions (or methods) that test some functionality for your program (what attributes of your object can be tested for, what methods can be tested, and what guarantees do you have about how this object will behave?)
(this may require a bit more research than we cover in class so far! Just be sure to document your attempts and include that documentation in your project.)

    I have one function in place that prints an invoice containing all the attributes of my object.

6. Get another person's menu and remember to **keep notes** on the following:

   I used "project_1_redux.py" from the google drive folder.

   1. **Attempt** to add file reading and writing to their program, note whether you accomplish this or not, how difficult was this?

      I was able to add file reading and writing to their program, but only for the driver and make, model and year of the car. Everything else looked a bit too convoluted to attempt.

   2. Identify sections of their code that you think could or should be made into methods or functions.

      There are tons of functions in this code, so why is the main function so long? I think rather than having three functions to handle user input and one massive try and except clause in the main function, the main function should be broken down into functions for each attribute. I don't think I need to be asked if I'm happy with my current input every single time I input something.

      Everything seems to work well, but it just took me a while with the debugger to see how everything works. Massive blocks of text with no supporting comments are hard to work with.

   3. What tests would use to show that their programming is working correctly (or would even break their code)?

      When I enter "n" in the "Are you happy with your selection" prompt, I get stuck in a loop until I choose to go back to the main menu or input "y". Incorrect input seems to be handled well.

   4. If you have these installed, then when you run pydocstyle and pycodestyle on their code, what warnings and errors do you get?

      Line too long & missing docstring.

      1. What do these messages mean?

         The lines of code are too long, and the functions have not been defined.

      2. In your opinion are the programmer's style differences from the PEP standards acceptable?

         I don't think I have an opinion on line length, but docstrings would definitely be nice to have.