For this project, I used four files: A driver, a function container (module) and test file, for #3, and a simpler example program for problem 2 which is seen below..

Text

Description automatically generated

First, we can see 2a defined as I define the Vehicle class, which is closely followed by 2b where I list a myriad of attributes. We follow this with adding behaviors to satisfy 2c.

Then, we begin to create instances of the class, broken up by demonstrating access to our attributes by way of class behaviors to build user familiarity with them, and to satisfy 2f and 2g.Graphical user interface, text

Description automatically generated

This is then continued as we create 2 more class instances, and check/prove their difference from one another (excluding the wheels, to show that not *everything* has to be different for two class instances to be different from one another.

Text

Description automatically generated

Here, we can see the driver file for #3, which I had a blast writing (though I encountered some difficult bugs along the way) as objects allow for much more complex programming and thinking without becoming impossibly difficult to think about. Text

Description automatically generated

Text

Description automatically generated

And here are screenshots of the function module file that the problem 3 driver ran from. I found this particular section very satisfying and rather beautiful to code, as the similarity in the architectures makes for satisfying repeating waves of code.Text

Description automatically generated

Text

Description automatically generated

To satisfy 4, I made this test file, which is callable from the driver file, though, as I wrote this code after the driver file, I selected not to directly run this through the driver file itself. This file checks the default values of the two objects in this sandbox, vehicles and passengers, and reports summatively if any of the outputs are incorrect. For more information on 4c (simulating object interaction), please see the page on the driver file for #3, where that simulation is performed.

Shape, rectangle

Description automatically generated