Python Object Attributes Lab

Introduction

In this lab, we will practice defining classes and instance methods. We will also practice working with getter and setter (read and write) methods using properties and decorators to operate on instance variables.

Objectives

- Practice defining classes and instantiating instances of those classes
- Practice defining instance methods

Defining Classes and Instance Methods

In the cells below define a `Driver` class and define a `Passenger` class.

Our driver instance objects should have instance variables for first name, last name, miles driven, and rating. We can name these instance variables `_first`, `_last` (short for first and last name), `_miles_driven`, and `_rating`. We will want to be able to access, change, and delete these values using the appropriate properties.

After defining the above instance methods, define an instance method called `greet_passenger`, which returns the string `Hello! I'll be your driver today. My name is` followed by that driver's first name and last name (i.e. `Hello! I'll be your driver today. My name is John Doe`).

In [ ]:

```
# Define Driver Class here with properties for each instance variable
class Driver:

    def get_first(self):
        return self._first

    def set_first(self, first):
        self._first = first

    def del_first(self):
        del self._first

    first = property(get_first, set_first, del_first)

    def get_last(self):
        return self._last

    def set_last(self, last):
        self._last = last

    def del_last(self):
        del self._last

    last = property(get_last, set_last, del_last)

    def get_rating(self):
        return self._rating

    def set_rating(self, rating):
        self._rating = rating

    def del_rating(self):
        del self._rating

    rating = property(get_rating, set_rating, del_rating)

    def get_miles_driven(self):
        return self._miles_driven

    def set_miles_driven(self, miles_driven):
        self._miles_driven = miles_driven

    def del_miles_driven(self):
        del self._miles_driven

    miles_driven = property(get_miles_driven, set_miles_driven, del_miles_driven)

    def fullname(self):
        return "{} {}".format(self.first, self.last)

    def greet_passenger(self):
        return "Hello! I'll be your driver today. My name is {}".format(self.fullname())
```

In [ ]:

```
driver = Driver()
driver.first = "Rachel"
driver.last = "Jensen"
driver.miles_driven = 100
driver.rating = 4.9
print(driver.first) # "Rachel"
print(driver.last) # "Jensen"
print(driver.miles_driven) # 100
print(driver.rating) # 4.9
driver.greet_passenger() # Hello! I'll be your driver today. My name is Rachel Jensen
```

In the `Passenger` class, we will want our passenger instance objects to have the attributes first name, last name, and email. Let's continue using the leading underscore naming convention we employed in our `Driver` class and name these instance variables `_first`, `_last`, and `_email`. Define the appropriate instance methods using property and the appropriate decorators for reading (getting), writing (setting), and deleting instance variables.

Next, we want to define an instance method called `yell_name` which returns a string with the passengers name in all caps (i.e. "RON BURGUNDY").

In [ ]:

```
# Define Passenger Class here with properties for each instance variable
class Passenger:

    def get_first(self):
        return self._first

    def set_first(self, first):
        self._first = first

    def del_first(self):
        del self._first

    first = property(get_first, set_first, del_first)

    def get_last(self):
        return self._last

    def set_last(self, last):
        self._last = last

    def del_last(self):
        del self._last

    last = property(get_last, set_last, del_last)

    def get_email(self):
        return self._email

    def set_email(self, email):
        self._email = email

    def del_email(self):
        del self._email

    email = property(get_email, set_email, del_email)

    def fullname(self):
        return "{} {}".format(self.first, self.last)

    def yell_name(self):
        return "{}".format(self.fullname().upper())
```

In [ ]:

```
passenger = Passenger()
passenger.first = "Ron"
passenger.last = "Burgundy"
passenger.email = "ron.burgundy1984@gmail.com"
print(passenger.first) # "Ron"
print(passenger.last) # "Burgundy"
print(passenger.email) # "ron.burgundy1984@gmail.com"
passenger.yell_name() # "RON BURGUNDY"
```

Great work!

Summary

In this lab, we practiced defining classes, instance methods, and utilizing the property function to create, getter setter, and deleter properties so we could access and operate on our instance variables.