Assignment 12 – Objects

Learning Objectives

1. Use python syntax to define and use objects
2. Understand the difference between getter, setter, and mutator methods
3. Write magic methods

Evaluation

1. Are functions documented?
2. Do functions only deal with information passed via parameters?
3. Do functions only pass information back via return statements?
4. Do your comments contain your test cases and expected results?
5. Does your program use a main function?
6. Does your program satisfy all of the requirements listed below?
7. Does the program use classes?

Description

1. Create one python source file that a class to represent airline flights with the following methods
    a. An __init__ constructor that takes the parameters departure, destination, and distance and sets fields appropriate for these parameters (i.e., “my…” replacing “...” with the parameter name). The parameters have the following meanings:
        i. Departure: city of departure
        ii. Destination: destination city
        iii. Distance: distance between departure and destination city.
    b. Three getter methods for each of the three fields mydeparture, mydestination, mydistance.
    c. A __str__ method that returns a string that nicely summarizes the flight information (the three fields of the flight).
    d. A method called traveltime that takes a parameter, speed, which is the speed of the flight and computes the travel time from the source to destination. The formula to compute travel time is mydistance/speed.
    e. A method called “extend” that takes a parameter which is itself a flight object. This method returns None if the departure flight of the parameter flight object doesn’t match the destination flight of the self object. If the two things do match, the method returns a new flight object with self’s departure, the parameter’s destination, and a distance equal to self’s distance and the parameter’s distance added together.
    f. Write a main function that tests the above functionality as follows:
        i. Ask the user to enter information for two flights.

ii. Generate and store a third flight by calling the extend method on the first flight with the second flight as a parameter.

iii. Test each of the getter methods on the original two flights.

iv. Prompt the user for a speed value.

v. Test the traveltime method on each of the original two flights using the speed value the user entered as a parameter.

vi. If the combined flight (returned by the extend method) is not None, test the getters and traveltime methods on the combined flight.

vii. If the combined flight is None, print a message indicating that the combined flight could not be computed.

2. Use canvas to submit your one python file that solves steps a-f. Make sure your assignment solves all these steps.