

School Nova Computer Science 202
Homework 5 (due 10/31/2020)

Save your code as lastname_homework5.py and submit on Google Classroom

The primary goal of this homework is to focus on OOP and, specifically, working with two interacting classes.

Task 0

Review the previous class notes. This is especially important if you could not complete some of the tasks from the previous homework.

Task 1

We are going to create a simple model of an airline, which consists of two types of objects: airplanes and airports.

Create a class PLANE. Each plane should have the following attributes: (1) unique 5-digit ID passed by the user, (2) status (in air or on the ground), initially zero, and (3) airport assignment (which will be ID of an airport to which the plane is currently assigned), initially zero.

Task 2

Create a class AIRPORT. Each airport should have the following attributes: (1) unique 3-digit ID passed by the user, (2) location (an (x, y) tuple), passed by the user; and (3) a list of planes currently assigned to the airport, initially an empty list.

Task 3

Generate 3 airports and 5 planes. Do not assign your planes to airports just yet. Try to use random.sample() to avoid duplicate id's for your planes, airports, and airports' locations.

For the airport locations, assume that the whole world is a 10 by 10 grid (that is, there are 100 possible locations). Hint: you can use random.sample to pick random (x, y) tuples from a list of all possible tuples (locations).

Verify that you successfully created your airports and planes (it's up to you how to do that).

Task 4

Add a method to your PLANES class that randomly assigns the plane to an airport. Make sure to update the corresponding attribute of the airport (the list of planes current assigned to the

airport). Add this method to your planes initialization (def `__init__`). This way, when each plane is first generated it is automatically assigned to a random airport. Verify that your code works as expected.

Task 5

Add a method to your AIRPORT class that displays the basic information and current status of the airport. It should look something like this:

```
Airport 22. Location: (4, 1)
List of planes on the ground: [9762, 5918, 633]
```

Verify your code.

Task 6

Add a method to your PLANE class that displays the basic information and current status of the plane. It should look something like this:

```
Plane 5918 on the ground at airport #22
The airport #22 location is (4, 1)
```

Notice that the displayed information uses data from both classes (the plane id and airport assignment are attributes of the plane class while the airport location is an attribute of the airport class).

Verify your code.

Task 7* (optional challenge, extra difficulty)

Create a method for the airport class which identifies another airport which is closest geographically (based on the location coordinates (x, y)).

Verify your code.