

Python 2nd Project

Introduction to OOP programming

HosseinKoofi.ir

Instructions

- Before you begin, it is important that you read this entire document for instructions and advice.
- If you are unsure how to interpret the programming task, write your assumptions in a WhatsApp message.
- You may send your code to me as many times as you want, I will check and let you know if I find any issue.
- It is absolutely critical that you submit .py files that compile, make a zip from your files and send it to me.
- If there is a part of your code that does not compile, don't include it in your submission.
- You can use any built-in modules or PyPi modules to solve these tasks (Only if you think it is necessary).
- Your code needs to be general. That means it needs to work with any input numbers, not just with the given example numbers in the Tester file. **Don't hardcode!**
- Write your full name as a comment in each file!
- Download the zipped project file and import it in your IDE.
- **Important:** None of the methods that you define can contain any print statements.

Good luck and may you do your best!

Programming Task

In this project you are required to code Python classes and methods. Only a tester Python file and its expected console output are given to you. The tester file illustrates how instances of classes are supposed to be created, and how methods may be invoked on their instances. **Don't modify the tester!** From this tester file, which does not compile to begin with, you are required to:

1. Identify the missing classes and methods.
2. Create all missing classes, as well as add headers of all required methods (i.e., name, input parameters).

Completing the above two tasks should make everything compile.

3. Implement the required methods (with any extra attributes or helper methods which you consider necessary), such that executing the given tester produces the expected console output.

System Requirements

- 0) Code compiles.
- 1) You need to develop a flight management system for an airline. Each flight is characterized by its flight number, origin airport, destination airport and distance. Airports are characterized by their three-letter airport code and country code. Airplanes have a type name, range in km and number of seats. Passengers have a name and integer passport number.
- 2) Each flight stores a list of passengers.
- 3) Passengers store a list of flights they are taking. When a passenger is added to a flight, this flight also needs to be added to the list of flights stored by the passenger.
- 4) Each flight has an airplane type. Before a plane is added, the method needs to check whether the plane's range is sufficiently large for the flight distance. If successful, the method needs to return true. If unsuccessful, it needs to return false.
- 5) A method is needed to determine whether the flight is overbooked i.e. it checks whether there are more passengers on the flight than seats on the plane. If the number of passengers is larger than the number of seats, the method needs to return by how many passengers the flight is overbooked. If not, the method needs to return zero.

- 6) Flights can be domestic (i.e. origin and destination have the same country code) or international (i.e. origin and destination have different country codes). We need a Boolean method that determines whether a flight is international or domestic.
- 7) The airline needs a list of any passengers on a particular flight who have not yet entered their passport numbers, which is signified by a passport number of 0 in the system.
- 8) Passengers receive frequent flyer points for every flight they take. 1 point for domestic flights, 3 points for international flights. Based on a passenger's list of flights, the system needs to calculate their total number of points.

You are required to write, valid Python syntax, classes, attributes, and methods to implement the above system requirements. Study the FlightTester Python file and its expected output carefully. It indicates the classes and headers of methods that you need to define. You are forbidden to define additional classes, whereas you are free to declare extra attributes or helper methods as you find necessary.