CS 115 - Introduction to Programming in Python

Lab Guide 07

Lab Objectives: Inheritance

Notes:

a) Upload your solutions as **a single .zip file** to the Lab07 assignment for your section on Moodle. You must use the following naming convention: Lab07_Surname_FirstName.zip where Surname is your family name and FirstName is your first name.
b) Solutions sent through email will not be accepted.
c) You should only use functionality covered in CS115 in your solution.
d) Include a docstring for your functions.

You will write a program for a cab owner to store and display information about their taxi cabs.

1. Create a class, `Cab`, with the following data attributes and methods. Note all data attributes and class variables should be private.

Class Cab:

Data Members:

- `typeOfCab`: private attribute that stores the type of cab, hatch back or sedan.
- `kms`: private attribute that stores the number of kilometers travelled.
- `year`: private attribute that stores the year the cab is produced

Methods:

- `__init__()`: initializes the data members to the values passed as parameters.
- Get methods for kms, type, year.
- `__gt__()`: compares Cab objects by their year if their types are the same.
- `__eq__()`: returns True if two Cabs have the same year and type, False if not.
- `__repr__()`: returns a string representation of a Cab object. See sample run for details.

2. Create a subclass, `Sedan`, by extending the superclass `Cab`, with the following data attributes and methods. Note all data attributes should be private.

   Data Members:
   - `price_per_km`: private class attribute (not instance) that stores the price per km ($2.5).

   Methods:
   - `__init__()`: initializes the inherited data members to the values passed as parameters.
   - `calculate_fare()`: calculates and returns the cab fare using the price per km and the number of kms.

3. Create a subclass, ***`Hatchback`***, by extending the superclass *Cab***,** with the following data attributes and methods. Note all data attributes should be private.

   Data Members:
   - `price_per_km`: private class attribute (not instance) that stores the price per km ($2.2).

   Methods:
   - `__init__()`: initializes the inherited data members to the values passed as parameters.
   - `calculate_fare()`: calculates and returns the cab fare using the price per km and the number of kms.

4. Write a script `CabApp` with the following functions:

- `find_greater()`: Takes a list of Cabs and a Cab object `cab` as parameters. The function should find and return the number of Cabs in the list with the same type as `cab` and whose number of kilometers is more than the kilometres of the `cab` passed as a parameter.

- `read_file()`: Takes a filename as a parameter. Assume each line of the file contains the type of cab and the number of kilometers, the year produced separated by a semicolon. Examine `cabs.txt` file. Using data in the file, return a list of Cab objects (Sedan or Hatchback).

The script should do the following:

- Creates a list containing Cabs using data in the file, *`cabs.txt`*.
- Display the fare of each Sedan cab in the list.
- Find and display the number of Sedans newer than the year 2015. Use the *`find_greater`* function.
- Find and display the total number of kilometers travelled by all Hatchback cabs with the year 2020.

Sample Run:

```
Sedan 1 will pay 500.0 TL
Sedan 2 will pay 1350.0 TL
Sedan 3 will pay 500.0 TL
Sedan 4 will pay 50.0 TL
Sedan 5 will pay 25.0 TL
Sedan 6 will pay 500.0 TL
Sedan 7 will pay 500.0 TL
Sedan 8 will pay 50.0 TL
Sedan 9 will pay 250.0 TL
Sedan 10 will pay 50.0 TL
Sedan 11 will pay 250.0 TL
Sedan 12 will pay 125.0 TL

There are 7 Sedan cars newer than 2015

All Hatchback cars of year 2020 have travelled 150 kms
```