BLG252E Object Oriented Programming Assignment 2

Release Date: April 8, 2020
Due Date: May 04, 2020, 23.59

Late submissions will NOT be accepted. Please submit your homeworks only through Ninova. This is an individual homework and in case of inappropriate exchange of information is detected, disciplinary actions will be taken.

- Your code must be compiled with following command: g++ 150150113.cpp
- Write your name and ID on the top of each document that you will upload.
- Compile your code on Linux using g++. You can test your code through ssh. Do not use any pre-compiled header files or STL commands. Be sure that you have included all of your header files.
- Use comments to explain your code.
- Be careful with the methods/attributes which are supposed to be constant, static, private.
- Handle exceptions and display error messages wherever necessary.
- If you have any questions about the homework please contact me via inceoglua@itu.edu.tr
- Submit your homework in a .zip file where it contains all of the necessary .h and .cpp files.

Problem Description

In this homework you are going to implement a shipping business management utility.

Part 1:

- You need to implement a **Person** class.
    - A person is represented with following attributes:
        - name, surname
- You need to implement an **Owner** class as a *subclass* of **Person** class**.**
    - The class should have an attribute named “ownership”, which represents the amount of ownership as percentage.
- You need to implement a **Courier** class as a *subclass* of **Person** class.
    - A courier has an attribute named “vehicle”, which represents the vehicle type for carrying shipments.
        - There can be 3 different types of vehicles which are: car, motorcycle, and bicycle. If any other type is given, an error message should be printed.

- In addition, a courier has a transportation capacity depending on the vehicle being used (i.e., 10 kg for bicycle, 35 kg for motorcycle, and 200 kg for car).

Part 2:
In the second part you are required to implement a Business class.

- A business is represented with name and address.
- A business can have multiple owners. If there are more than one owner, ownerships are distributed equally (e.g. for 2 owners 50%-50% etc.). The owners are fixed and they are provided during construction as an array. The amount of "ownership" should be adjusted during construction as well.
- A business can have any number of couriers. New couriers can be hired or existing ones can be fired.
- You can define extra variables if needed (e.g. counters).
- The Business class should have following functionalities:
  - hire_courier: This method accepts a Courier object as argument and inserts to courier list/array.
  - fire_courier: This method accepts a Courier object, searches the courier and removes it from the list/array. If the courier does not exist it should print an error message. You should overload == operator for Courier class which is required for the search part.
  - list_couriers: This method prints all information (name, surname, vehicle_type) about all couriers to screen.
  - list_owners: This method prints all information (name, surname, ownership) about all owners to screen.
  - calculate_shipment_capacity: This method should sum up total transportation capacity.
  - Overload () operator: When () operator is invoked, it should print all information about the business (i.e., name, address, owners, couriers). You can use list_couriers and list_owners methods that you have implemented.
  - Overload [] operator: When [] operator is invoked, it should return courier with the given index. If the index is invalid, it should print an error message.

Notes:
- If you use arrays, you should allocate memories dynamically.
- Make sure your code does not have any memory leaks.
- You should implement required Constructors, Destructors, getters and setters.
- Make use of constants, call by reference appropriately.
- Sample main program is provided. Your codes will be tested with other inputs as well.

Output for the provided sample main program:

```
Owner_name1 surname1 50
Owner_name1 surname1 50
Atlas Maslak Istanbul/Turkey
Owner_name1 surname1 50
Owner_name1 surname1 50
Courier surname1 car
Courier surname2 motorcycle
Courier surname3 motorcycle
Courier surname1 car
Courier surname3 motorcycle
235
```