CS 115 - Introduction to Programming in Python
Lab Guide 8

Lab Objectives: Classes and Inheritance.

a) Create a class, **`Employee`**, which has the following data members and methods. Note all data members and class variables should be private (__).

Data Members:

- **`taxRate`**: class data member, same for all Employees.
- **`empName`**: string name of the Employee.
- **`idNum`**: numeric identification number of the Employee.
- **`wage`**: the float wage of the Employee.

Methods:

- **`__init__():`**
  - takes a employee name, id number and wage, and using the set methods, sets the attributes to the given values.
- Get methods for all data members.
- Set methods for all data members implemented according to the following:
  - Employee name must consist of only alphabetic or space characters. If name is invalid, empName should be set to `'default'`.
  - Id number should be converted to an integer value. If the id number cannot be converted to an integer, set to the default, 99999.
  - Wage must be a positive value. If the value is negative, set the wage to zero.

- `calculate_salary()`**:** returns the salary, which is the wage minus the tax (using `taxRate` value)
- **`__lt__():`** an Employee is less than another if their surname is less. If the surnames are equal, compares the alphabetic first names. Note: You may assume that all Employees have at least one name, and one surname. The surname will always be after the final space in the name.
- **`__eq__():`** Employee objects are equal if their id numbers are equal.
- **`__repr__():`** returns a string representation of an Employee object. See the sample run for formatting.

.

b) Create a class, **`PartTime`**, which is a subclass of Employee. A PartTime Employee has the following extended data members and methods. Note all data members and class variables should be private (__).

Data Members:

- **`hours`**: the int hours worked of the Employee.

Methods:

- **`__init__():`**
  - takes the employee name, id number, wage, and hours. Calls the Employee init, to initialize the inherited data members, and sets the hours to the value passed as a parameter.
- Get method for `hours`.
- `calculate_salary()`**:** overrides the inherited method. Returns the salary (which is the wage multiplied by the number of hours) minus the tax (using `taxRate` value)
- **`__repr__():`** returns a string representation of a PartTime object. See the sample run for formatting.

c) Write an application, **`yourname_Lab08.py`** that does the following:

- Creates 2 Employees and 2 PartTime objects and adds all to a list.
- Print the list of Employees.
- Sort the list of Employees.
- Print the sorted list of Employees.
- Display the calculated salary of all Employees.
- Display the average calculated salary for PartTime Employees.

Sample Run:

```
Original List:
[
Name: Evren Kilic Employee ID: 51642 Wage: 275000,
Name: Zana Zengin Employee ID: 12345 Wage: 650 Hours: 45,
Name: Ayse Kilic Employee ID: 98765 Wage: 315000,
Name: Anisa Nalan Employee ID: 64276 Wage: 475 Hours: 20]

Sorted List:
[
Name: Ayse Kilic Employee ID: 98765 Wage: 315000,
Name: Evren Kilic Employee ID: 51642 Wage: 275000,
Name: Anisa Nalan Employee ID: 64276 Wage: 475 Hours: 20,
Name: Zana Zengin Employee ID: 12345 Wage: 650 Hours: 45]

Salary of all employees:
Ayse Kilic salary after tax: 189000.0
Evren Kilic salary after tax: 165000.0
Anisa Nalan salary after tax: 5700.0
Zana Zengin salary after tax: 17550.0

Average Salary for PartTime Employees: 11625.0
```