CS 115 - Introduction to Programming in Python

Lab Guide 7

Lab Objectives: Classes and Objects. Inheritance.

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

Data Members:

- **`dname`**: stores the name of the Doctor.
- **`title`**: stores the title of the Doctor (Professor, Associate Professor, Assistant Professor, Specialist)

Methods:

- **`__init()__:`** initializes the name and type to values passed as parameters.
- **Get and Set methods for:** dname and title.
- **`__eq__` :** attendee object are equal if their name and title are the same.
- **`__lt__` :** an Doctor is less than another based on their alphabetic titles, if the titles are the same it then compares according to alphabetic name.
- **`repr()__:`** returns a string representation of a Doctor object. Format: `'Title – Doctor Name'`

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

Data Members:

- **`patients`**: stores the integer number of patients seen by the Doctor.
- **`treatmentFee`**: stores the fee per patient for a treatment.

Methods:

- **`__init()__:`** initializes the dname and title by calling the super class init method. Initializes the number of patients and treatment fee to values passed as parameters.
- **`calculate_payment` :** calculates and returns the payment, which is the number of patients multiplied by the treatment fee.
- **`repr()__:`** returns a string representation of a Private object. Format: `'Title – Doctor Name Payment: payment\n'`

c) Create a class, **`State`**, which is a subclass of Doctor. The subclass has the following data members and methods. Note all data members and class variables should be private (__).

Data Members:

- **`salary`**: stores the monthly salary of the State Doctor.
- **`baseBonus`**: the same for all State Doctors, defined as a static class member, not inside the init method. Constant value 5000.

Methods:

- **`__init()__:`** initializes the dname and title by calling the super class init method. Initializes the salary to the value passed as parameters.
- **`calculate_payment :`** calculates and returns the payment, according to the following: Professors receive their salary plus 125% of the baseBonus, Associate Professors receive their salary plus 100% of the baseBonus, AssistantProfessors and Specialists receive their salary plus 75% of the baseBonus.
- **`repr()__:`** returns a string representation of a State object. Format: `'Title – Doctor Name Payment: payment\n'`

d) Write an application, **`Lab07.py`** that does the following:
- Reads the data from the file doctor_data.txt. Using the data from the file, creates Doctor objects (State or Private) and if the Doctor is not already in the list, adds the Doctor to a list.
- Sorts the list of Doctors.
- Displays the sorted list of Doctors.

Sample Run:

```
Professor - Alyssa Padilla Payment:150000.0
 not added, duplicate!
[Assistant Professor - Alonzo Ballard Payment:9250.0
, Assistant Professor - Tracey Russell Payment:90000.0
, Associate Professor - Andrea Howard Payment:20000.0
, Associate Professor - Rosalie West Payment:35000.0
, Associate Professor - Sue Beck Payment:139500.0
, Professor - Alyssa Padilla Payment:150000.0
, Professor - Darryl Walker Payment:100000.0
, Professor - Jeremiah Bailey Payment:24750.0
, Specialist - Andrew Austin Payment:12750.0
, Specialist - Lyle Romero Payment:10250.0
```