INFO I211 - Information Infrastructure II

Spring 2014

Lab 5: Python classes and private attributes.

How to work on your lab task - how to turn in your program:

Turn in your lab 5 task as "20140214 Lab 5 task" in the I211 Oncourse site -> Assignments thus:

- You will work on this task with your lab team, but only *one* team member needs to turn in the Python solution code *while in lab*, to receive full credit for the work done.
- For today's lab, the person responsible to turn in the Python solution code is the *first* person listed in your lab team, as from the `teams-lab...` file on Oncourse. If the first person is absent from lab, the second person on the list is responsible to turn in the Python solution code, etc.
- Name your file as `YourTeamNumber_YourUsername_Lab05.py`
  (e.g. If I were to upload, the file would be named `0_mitja_Lab05.py`)

  Upload your program file (.py) to Oncourse under I211 Assignments->"20140214 Lab 5 task" as a .py file.
  Include the following information as a `#comment` at the top of the file you submit:

  ```
  # I211 Spring 2014 - Lab 5
  # your name (First, Last)
  # your IU email address
  # your I211 lab team number
  # the names of all your I211 team members
  ```

  Kindly make sure that your program successfully executes in Python 2.x before submitting, without syntax errors. Test your code on a wide range of data.
  To receive any credit for the work done, the Python source code file has to be turned in *at lab time and while in lab*. (including the information about themselves and their student team, as listed above. Submissions lacking this information will lose 50% of the assigned grade.)

Tasks

1. Design and implement a Python 2.6 class to keep track of customer support requests, for services related to musical instruments repairs and tuning.
   Each object in this class should contain these attributes - make all these attributes *private*, but you don't have to turn them into *properties*:
   - Customer name (a string in the format "LastName, FirstName")
   - Instrument serial number (a string which may contains letters A-Z and digits 0-9).
   - Type of request (a string description such as "String Replacement", "Woodwind Repad", etc.
   - Date of service (a number in the format YYYYMMDD)
2. The class should also contain these *class attributes*:
   - The customer name for the *most recent support request*, i.e. the last instantiated object in time order. (a string in the format "LastName, FirstName")
   - Total number of customers. (an integer number)
   - Total number of instruments ever serviced -- since any customer may have multiple instruments. (an integer number)
     (*Note*: these last two have to be tracked separately, and may or may not have the same value, depending on the instantiated data).

3. The class should provide these *methods*:
    - A class method to find the most recent support request.
    - Object methods to find out the customer name, the instrument serial number, the type of request and the date of service.
4. In your class, do *not* rely on *printing* methods such as `__str__`.
5. In your main Python program, instantiate three objects representing three separate customer support requests.
   Hint: each instantiation should also correctly update all class attributes - this can be done in the *constructor* method.
6. In your main Python program, print the the customer name, the instrument serial number, the type of request and the date of service for *one customer support request*. Obtain these attributes using the object methods you designed.

Last updated: *Mitja Hmeljak* February 13, 2014