Lab 8

Objectives:

Practice creating your own classes and using them in your programs.

1. Implement a Person class.
Use the following comments as guidelines (copy and paste them into a new Java file for the Person class and fill in the required Java code).

   // instance variables: name, age, and happiness(true or false)

   // constructor: everyone starts out at age 0
   //   and happy (happiness = true)
   // Note: needs one parameter only, for the name.

   // birthday(): increases age by one

   // makeHappy(): causes this person to become happy
   // (sets the instance variable for happiness to true)

   // makeSad(): causes this person to become not happy

   // getAge(): returns the age of this person

   // toString(): returns a String describing this person.
   // the description includes the name and age and the words

2. Implement the driver class
You can call this class PeopleBeingPeople or another name or your choice. Use the following comments as guidelines (copy and paste them into a new Java file and fill in the required Java code – be sure to set up the main() method appropriately).

   // Instantiate three objects of the Person class, assign them
   // to variables named friend1, friend2, friend3, respectively.
   // (Use names of your choice.)

   // Print out info about friend1, friend2, friend3.

   // use the makeSad() method to make each of them unhappy.
Lab 8

// Print out info about friend1, friend2, friend3.

// use the birthday() method several times to increase all the ages.

// use the makeHappy() method to make friend1 happy.

// Print out info about friend1, friend2, friend3.

3. Add another method to the Person class, speak()
If the person is happy, the method should return the String “I have nothing to complain about”, otherwise, the String “I am so sad”.
NOTE: The method should NOT print anything, it should just return a String with the appropriate message.
Add appropriate code to the driver class to test this method.

4. Add another method to the Person class, starz()
This method should return a String that consists of N asterisks, where N is the age of the Person. For example, if the Person’s age is currently 5, it should return the String: "*****"
Add appropriate code in your driver class to test this method.
Hint: You need to use a loop and be careful to ensure that starz() does not actually print anything, it just returns a String, which the driver class can print.

5. Now we will add accessor and mutator methods (i.e., “setters” and “getters”):
• For the instance variable name in the Person class, these methods should be called
  getName() and setName() respectively.
• Since we already have other ways to set the age and happiness of a Person object,
  we only need to implement accessor methods for these instance variables (we don’t need
  mutators):
  o the accessor method for age should be called getAge()
  o the accessor method for happiness should be called isHappy()

    ▪ we name this accessor isHappy() instead of getHappy() because it returns a
      boolean value (and also because “getHappy()” might be misleading!).
    ▪ This is a common naming convention for methods that return a boolean value,
      since it sounds more natural in a statement like this:

      if (friend1.isHappy() && friend2.isHappy() && friend3.isHappy())
        System.out.println("All my friends are happy, so I am happy.");
      else
        System.out.println("You cannot make everybody happy.");

Add appropriate code in your driver class to test these new methods (including code such as
the if/else above).
Lab 8

6. *(Optional)* In the **Person** class, add another instance variable, to represent a **picture** of the person.

This instance variable should be of the **ImageIcon** class. (Refer to Lab 7 if you need a refresher on how to set up an **ImageIcon** object.)

Modify your constructor so that it initially sets the image of the **Person** to this:

![Photo Not Available](http://www.csc.villanova.edu/~map/1051/images/photo%20not%20available.jpeg)

7. *(Optional)* Add mutator and accessor methods for **picture**

Add appropriate code in your driver class to test these methods. You can use the images below or images of your choice:

![Rugrats Images](http://www.csc.villanova.edu/~map/1051/images/rugrats%20anjelica.jpeg)
![Rugrats Images](http://www.csc.villanova.edu/~map/1051/images/rugrats%20tommy.jpeg)
![Rugrats Images](http://www.csc.villanova.edu/~map/1051/images/rugrats%20chuckie.jpeg)

http://www.csc.villanova.edu/~map/1051/images/photo%20not%20available.jpeg

**NOTE:** You will not be able to display the images yet, because your program so far does not have a GUI. The program can still store them though, as part of the **Person** objects.

8. *(Optional)* Add some more code to your driver class to create a Graphical User Interface (GUI).

Your GUI should consists of a frame (use **JFrame** of size 1200x800) with a primary panel and subpanels for each of the people (use **JPanel**).

Each subpanel should contain two labels (use **JLabel**):

a) a label with the picture

• use the getter method `getPicture()` to obtain this and put it in the label

b) the description

• use `toString()` method to obtain this and put it in the label
9. Sketch a UML class diagram for your classes. Include all the instance variables and methods from previous questions (if you did the optional GUI-related questions, be sure to include the data and methods related to these).

**Note:** this lab will be checked in class. You do NOT need to hand in anything. Just be sure to demonstrate your work – your program and the UML class diagram.