Lab 8  Name:__________________________  Checked:_____

Objectives:
Practice creating classes and methods, and using them in your programs.

1. Starting with the Account class and OnePercent classes from the Account class exercise - see http://www.csc.villanova.edu/~map/1051/s15/04AccountExercise.html

a) Demonstrate your program and Account exercise worksheet:
Instructor/TA: ____________________________________________

Recall that Java allows you to define alternative versions of methods using the same method name as long as the different versions also have different a different number or types of parameters. In this exercise you will define alternative withdraw() and constructor methods.

b) Add another version of the withdraw() method. This version does NOT charge a withdrawal fee, so it has only one parameter. (Use this version of the method in OnePercent to withdraw the taxes from the accounts.

c) Add another version of the constructor that takes only 2 parameters: name and account number (ie, no initial balance). This constructor creates an Account object with initial balance $0. Modify OnePercent to use this version of the constructor to create the “Uncle Sam” account.

d) Create a new method that adds interest to the account, according to the rate given by its parameter. For example, if the acct1 balance is $100.00 and the method is invoked as follows: acct1.addInterest(0.015);
the balance of acct1 should increase by 1.5% (so $100 + $1.50 = $101.50 ). Test your method by invoking it four times to add interest to all the accounts (including Uncle Sam’s!).

2. Implement a Person class.

a) Copy and paste the Java comments below into a new Java file for a Person class (we will use these comments to build the code for the Person class incrementally).
b) Start by putting in the class heading and the enclosing braces; write the code for the instance variable declarations and implement the constructor and toString() method.
c) Compile your class and fix any errors before proceeding.

//*******************
// Person.java       Author: YOUR NAME HERE
// Represents a person, with attributes: name, age.
//*******************

// instance variables: name, age

// Constructor: Sets up the person by defining the name, and age

// toString():returns a String describing this person, eg: "Jasmine, 19"
2. Implement the client (driver class).
You can call this class PeopleBeingPeople or another name or your choice.
Use the comments below 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).

```
//**********************************
***********
********************
//  PeopleBeingPeople.java          Author: YOUR NAME HERE
//  Driver class to test Person class.
//********************************************************************************
//  main(): creates some Person objects, prints their info.

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

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

3. In the Person class, add another constructor that has only one parameter, the name. Modify your driver class to use this constructor to create an additional Person object friend4 and to print out info about friend4.

4. Let’s now add some more methods to the Person class.
Copy/paste the comments below into your Person class and fill in the code as appropriate.

```
//-----------------------------------------------------------------------------------------------
// birthday(): increases age by one.
//-----------------------------------------------------------------------------------------------

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

5. Test your methods by adding some code to your client (PeopleBeingPeople class).
Increase the age of friend4 twice and to then compute and print the average for the ages of
the four friends (i.e., use getAge() to obtain the ages of the four friends, add them together
and divide by four).

Demonstrate your work to the instructor/TA: ________________________________