

ComS 207: Programming I

Homework 4

Out: Wed. Sep 19, 2007

Due: Fri. Sep 28, 2007

1. Programming Projects

Choose **four** of the following **six** programming projects and implement them. Your grade will NOT depend on which ones you choose. Just pick the ones that you like.

(a) Cats

Design and implement a class called "Cat" that contains the following: 1) Name; 2) Cat Age; 3) Male or female cat; and 4) lifetime number of mice eaten. All of these variables should be declared *private*. In the Cat constructor, the name of the Cat should be specified. Include getter methods for 1-4 and setter methods for 2-4 above.

Create a method called `peopleYears()` that accepts a cat's human age and returns it in cat years (5.9 cat years per each human year). Next create four Cat objects with properties 1-4 above in a program called `CatInfo`. Get and set each property for all cats and print some meaningful messages such as:

```
System.out.println("Cat 1 is called " + cat1.getName());
System.out.println("Cat 2 has caught " + cat2.getMiceEaten() + " mice so far");
```

(b) Shopping cart

Create 3 shopping carts from a "Cart" class. Each class should have four items in it, a title, and a price for 4 items. Your shopping cart should use private variables, so use getter and setter methods to access the items (variables) in your Cart Object. After you have created three Cart objects with four items each, you should print the cart contents. Use a method called `displayCart()` to make it easier to print out the items for each shopping cart. A sample output is given below:

```
----- Your Shopping Cart -----
1. Paper           4.99
2. Pencils         1.49
3. Pens           2.39
4. Printer Ink     39.99
-----
Total:             48.86

//cart2 " " "
//cart3 " " "
```

(c) Team Roster

Create a program called TeamRoster that uses a class called Roster. Roster should contain a team name and the names of all the players on the team written into one string. Use a method called addTeamMember() to append a new member to your roster string. After creating two rosters with more than 5 members, print out the team names and their rosters as show below:

Sample Output:

Instructors

1. Alex
2. Dustin
3. Andy
4. Geoff
5. Jacob
6. Debra
7. Faraz

Total Members: 7

Students

1. Student 1
2. Student 2
3. Student 3
4. Student 4
5. Student 5

Total Members: 5

You should indicate the total number of members in each group near the bottom and label each person with an index (e.g. 1,2,3...).

(d) Spheres

Design and implement a class called Sphere that contains instance data that represents the sphere;s diameter. Define the Sphere constructor to accept and initialize the diameter, and include getter and setter methods for the diameter. Include methods that calculate and return the volume and surface area of the sphere (Volume= $\frac{4}{3}\pi r^3$, area = $4\pi r^2$). Include a toString method that returns a one-line description of the sphere. Create a driver class called MultiSphere, whose main method instantiates and updates several (at least 5) Sphere objects.

(e) Books

Design and implement a class called `Book` that contains instance data for the title, author, publisher, and copyright date. Define the `Book` constructor to accept and initialize this data. Include setter and getter methods for all instance data. Include a `toString` method that returns a nicely formatted multi-line description of the book. Create a driver class called `Bookshelf`, whose main method instantiates and updates several (at least 5) `Book` objects.

(f) Flights

Design and implement a class called `Flight` that represents an airline flight. It should contain instance data that represents the airline name, flight number, and the flight's origin and destination cities. Define the `Flight` constructor to accept and initialize all instance data. Include getter and setter methods for all instance data. Include a `toString` method that returns a one-line description of the flight. Create a driver class called `FlightTest`, whose main method instantiates and updates several (at least 5) `Flight` objects.

Please use data for some real flights. For example, you can get some of that from the web page for the Des Moines airport: <http://www.dsmairport.com/fids/ARR-TIME.HTM>

2. * For Advanced (or Bored) Students Only!

Develop an application that implements a prototype user interface for composing an e-mail message. The application should have text fields for the To, CC, and Bcc address lists and subject line, and one for the message body. Include a button labeled Send. When the Send Button is pushed, the program should print the contents of all fields to standard output using `println` statements.

3. What to Submit

For part 1 (and 2 if you chose to do it) submit your code electronically using WebCT.

IMPORTANT: Once again, no late homeworks will be accepted.

That's it. Good Luck!