CSE 142 - Summer 2003

Homework 2 - Due: Sunday, July 6 9:30 PM

A link to the turn-in page will appear on the course calendar webpage.

Written Assignment

These questions are meant to get you thinking some more about data types and class design so you'll be ready to implement them in the programming assignment below. There are also some basic Java comprehension questions. To gain full credit on each question, your answer must have no significant flaws, must be clear to the reader, and must be complete (but remember that complete does not mean verbose -- be concise and to the point).

As always, you should record your answers to the questions in a plain-text file (for example, "answers.txt") and then submit that text file along with the rest of your electronic turn-in. The turn-in page will tell you where each of your files should go. MS Word .doc files will *not* be accepted for grading.

1. Imagine that you're writing a program to play checkers.

   a) Suggest three objects you might use to model parts of the game.

   b) For each object, write down at least two properties and two responsibilities that it might have. For each property, suggest a data type that you could use to keep track of that kind of information (the same as we did in class).

2. Consider the following sequence of Java code. Pretend that you are the computer's processor, executing each line in order. (Sometimes it helps to jot down the variables' changing values in a table.)

```
/** growth rate */
double rate = 0.1;

/** base amount */
int base = 20;

/** cumulative amount */
double toDate;

/** interval since last update */
double interval = 10;

toDate = base;                          // (a)
toDate = toDate + (interval * rate);    // (b)
interval = interval + 15;               // (c)
toDate = toDate + (interval * rate);    // (d)
```

   a) What is the value of toDate after statement (a)?
   b) What is the value of toDate after statement (b)?
   c) What is the value of interval after statement (c)?
   d) What is the value of toDate after statement (d)?

3. This question requires you to use your web browser and the documentation for the Java class libraries. If you don't already have a bookmark in your browser that takes you directly to the JavaAPI documentation, you should set one up. The documentation is available online and for download, and the last link on the course web's "Java Libraries" page takes you to it.

   Go to the JavaAPI documentation web page. In the upper left corner of the screen is a set of "packages". Scroll down until you find "java.lang" and click on it. Now scroll the lower left frame down until you see the class name String. Click on that. You will then see a description of the String class in the main part of the window.

   Refer to the "Method Summary" section. (Questions are on next page.)

3a) There are two versions of the method "toLowerCase". What is the purpose of the version that takes no parameters? You can copy the text directly from the documentation page - you don't need to explain the details in your own words.

3b) What is the type of the value returned by the "length" method?

Programming Assignment

Turn in PetSet.java and Whatever.java (or whatever you decide to name your new class) when you have completed this task.

This project has you

a) develop one new class (Whatever) to describe a kind of pet other than a cat or a dog, and

b) complete another class (named PetSet) to add your new pet (Whatever) to the group of pets already known.

Completely implemented Cat, Dog, and PetSet classes are provided. You need to create your new Whatever class and update the implementation of the PetSet class to include it.

1. If you haven't done so already, download the cse142-hw2.zip file and unzip it. The project skeleton is in the directory hw2. Your task is to implement the two new classes.

2. Start DrJava, open the PetSet source file and read it. You can see that the function of a PetSet is to allow easy control of a bunch of pet objects. As provided to you, it does the job for one Cat and one Dog.

3. Open the Dog or Cat source file and read it. You can see that each animal has a few properties and a few responsibilities (queries and commands). Your task is to create another animal of your own design and write a class for it. The new class should be based closely on Cat.java and Dog.java, and should provide all the same functions, implemented as appropriate for your animal. This can be a Bird, a Snake, an Aibo, or whatever you want. This class does not need to be any fancier than Cat and Dog, just different. In the constructor that only takes one parameter your pet should set different default values.

4. Once you have your new animal working, go back and modify PetSet to include the new beastie in all the responsibilities implemented in PetSet. You need to update the PetSet constructor, and the snack, dine, and reportWeights methods.

5. Here is an example of the output generated by PetSet when the final program is run. My additional animal is a bird named Tweety in this example.

```
> java PetSet
Smoky weighs 5.0 pounds.
Jessica weighs 18.0 pounds.
Tweety weighs 0.2 pounds.
snacking ...
dining ...
Smoky weighs 5.075 pounds.
Jessica weighs 18.375 pounds.
Tweety weighs 0.23 pounds.
>
```