IST 256
Lab Week 13, Thursday, April 19, 2012

1. Designing a Class

Suppose that we have a file Pictures.txt that contains information about images. Each line of the file has the name of the image, the name of the photographer, the number of prints sold and the type (either Color or B&W for black and white). An example file might look like:

```
Mountain,Adams,40,B&W
Workers,Salgado,31,B&W
Flower,Mapplethorpe,25,Color
Baseball,Allen,200,Color
Woman,Lange,38,B&W
```

a. First design a class that is suitable to keep the data from each line of this file. The name of the class should be Picture and it should have fields for imagename, photographer, numprints, and medium. These fields should all be public and there should be one Constructor method that initializes the fields. No other methods are required at this time.

b. Write the declaration of a one-dimensional array of objects of the type Picture defined by the class in part a. The array should allow up to 12 elements.

b. Designing a SubClass (Optional Extra Credit Lab Problem)

Option 1:

```
/*
 *  Class to represent data about each student
 */
public class Student {
//   name, gender (either M or F), age in years, height in inches, email address
   protected String studentname, gender;
   protected int age, height;
   protected String email;

   // Constructor gives an initial value to all fields
   public Student(String _name, String _gender, int _age, int _height, String _email)
   {
      studentname = _name;  gender = _gender;  age = _age;
      height = _height;  email = _email;
   }

   // accessor methods
   public String getStudentname ()
   {
         return studentname;
   }

   // method to get contact information
   public String getContact ()
   {
         return email;
   }
}
```

The above class represents the data for a Student and has a field for an email address that is returned as the result of a method to getContact information.

Define a subclass of the Student class that is a LocalStudent. This class will have an additional field for a String that will contain a local phone number. Change the constructor function to also initialize this new field. Give a different method for getContact() that returns a String with both the email and the phone number put together into one String.

Write an answer for the LocalStudent class in the space given on the next page.

Option 2: Extend your StudentData project in NetBeans to have the LocalStudent class as a subclass.

In this case, you will also need to add to the Student class:

the email field and the getContact() method,
an additional no argument constructor:
public Student() { }
and change the private fields to be protected so that they can be accessed by the subclass

Then create the LocalStudent class as described above.

(Note that when you start to write the LocalStudent extends Student, you will get a red underlined error that is just NetBeans saying that you haven't completed it yet. Just ignore the red and continue to complete the class.)

This is extra space to write an answer for the LocalStudent option 1 problem. To complete Option 2 for the bonus credit, print the LocalStudent class and hand it in.

This lab is due on Tuesday, April 24.