IST 256
Lab Week 12, Monday, April 12, 2010

(Make sure your name gets on an attendance sheet.)

1. Review Using Arrays and Classes.

Suppose that we have a class that represents people with two fields for name and age.

```
public class Person
{
        // fields for the name and age
        private String personname;
        private int age;

        // constructor initializes all two fields
        public Person ( String startname, int startage)
        {
                personname = startname;
                age = startage;
        }

        // accessor functions for the name and age
        public String getName()
        {
                return name;
        }
        public int getAge()
        {
                return age;
        }
        ….
}
```

Suppose that we have a main method with an array declared of type Person, and that data is read into all 50 of the Persons in the array.

```
        Person [ ] personarray = new Person [ 50 ];
```

a. Write the first line of an If statement that will test if the name of the first Person in the array is “Alan”.

b. Write code that will compute the average of the ages of all 50 Persons in the array.

2. Extending the program with Student data from a file
In this example, we will again extend the TestStudent2 example that we did in the last two labs. Today we will add a button on the GUI so that we can search for students by name.

a. Start by **opening the TestStudent2** project (or whatever you called it).

b. Extend the form interface.
On the form, we will keep the previous buttons and labels for reading the file and saving the data, for displaying the students and for finding and displaying the tallest student. Now we will add a textfield to allow the user to type in a search name and another button to perform the search. The form can look something like this:

```
|__ Read Student File__|      File Status

|___Display Students___|                          |__Find Tallest Student__|
        Results:                                        Tallest Student:

Enter (partial) search name:  |___________|          (label and textfield)
        |__Search Student Names__|                        (button)
Results:                                                (multiline label)
```

When I did this example, I found that I had to explicitly make the Results label be multiline by dragging on the corners to make it bigger.
For the new button, select Event -> action -> actionPerformed.

c. Add Code for the new Search button
If you have completed the previous two labs, you should not have to modify your previous code, but just add code for this button.

This button will search the array of Students to find ones whose name matches according to the following criteria: the search string can be found as a substring anywhere in the student's name and case should be ignored. Find all the students that match.

- Declare a variable to hold the search name String and get the search name from the TextField.
- Declare and initialize two string variables to have the result header and the result names. The result names should be initialized to the empty string "".
- Write a for loop that iterates over all the students in the array up to numstudents. Compare each student name (using the getName accessor method) with the search name according to the criteria described above.
- If the student name matches, add that student to the resultnames.
- After the loop, if no students were found, add a No Match message to the resultnames.
- Display the resultheader and resultnames.
- **d. Test your program.**