

IST 256

Lab Week 12, Tuesday, April 12, 2011

1. Finish the program from the lab last week

In the last lab, we wrote a project for Students that had one button that would read a file of Student data and store it into an array of Students, and another button that went through the array and displayed every student. If you are not done with this yet, finish this lab, copy the declarations and button code to a text file and upload it to the lab dropbox in the iLMS.

2. Extending the program with Student data from a file

Today, we will extend the Students example with two more buttons. The first will display the tallest student, and the second will allow the user to search for a student by (partial) name.

a. Start by **opening the Students** 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, and for displaying the students. We will add a button to display the tallest student and a label to show (only) the name of 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__	__Display Tallest Student__
Students:	Tallest:
Enter (partial) search name: _____	(label and textfield)
__Search Student Names__	(button)
Results:	(multiline label)

For the new buttons, select Event -> action -> actionPerformed.

(You may find that you explicitly need to resize the Students label in the Design pane for all the results to show.)

c. Add accessor methods to the Student data class

In the new buttons, we will need to get the student name from each student class instance. So we should add an accessor method getStudentName to the class.

Go to the code for Student.java and add the following method where the comment says “//no accessor methods so far”.

```
public String getStudentName ()
{
    return studentname;
}
```

As usual, adjust the indentation, and change the comment to say “// accessor methods”

For the Tallest Student button, we will also need to get the height for each student, so we should also add an accessor method getHeight to the class.

Add the getHeight() method to the Student class following the example for getStudentName().

d. Add Code for the new Tallest Student button

This button will search the array of Students to find the index of the one which has the largest height in inches.

From the slides, adapt the code for finding the highest temperature to find the tallest student. The main difference will be that the highest temperature code uses the following comparison:

```
if (temperatures[i] > temperatures[indexSoFar])
```

and instead of comparing the value of each array element, each of our array elements is a Student and we want to compare the value of the height fields:

```
if (students[i].getHeight() > students[indexSoFar].getHeight())
```

Display the name of the student in the Tallest label.

e. Test your program.

Which student is displayed as the tallest? Is this the first or last of the tallest students?

f. Add Code for the new Search 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 starting substring of the student's name and case should be ignored.

Find and display 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 getStudentname 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 resulthead and resultnames.

g. Test your program.

Test your program with by typing in example search strings that match and don't match.

Write here examples that you tested:

1. An example that didn't match anything:

2. An example that matched one student:

3. An example that matched more than one student:

4. An example that shows the search is ignoring case:

This lab sheet will be due on Tuesday, April 19, at the beginning of class.