CS 180 Problem Solving and OO Programming
Fall 2011
Exam 2 Practice Problem Set
November 6, 2011

Note: *No solutions will be provided for these problems. Solve the problems and check out your solution in DrJava or Eclipse. Talk to the instructor or your recitation TA if you have questions. These problems will be discussed during the special class on Sunday Nov 6 4-7pm, LWSN 3102.*

1. Write a Java method named `setArray`(). This method is called as follows:

   int [][] diagonal=setArray(n);

   where n>0 is an integer. The `setArray`() method returns an `nxn` array of integers such that the value of the element located on the diagonal is the row number, and all the remaining elements of the array are set to 0. For example, if n=3 then the `setArray`() method will return the following array:

   ```
   0  0  0
   0  1  0
   0   0 2
   ```

2. Write a Java method named `dumpFile()` that takes the name of a file (a string) as input, opens this file, reads the data in the file, and prints the data read on the console-- line by line. The method is to be called as follows:

   `dumpFile`(name);// `name` is a string denoting the file name.

   The file contains one or more lines of data in the following format:

   ```
   LoginID grade
   ```

   where `LoginID` is a string with no spaces and `grade` is a real number less than 100 but greater than 0.

3. Modify the program in Problem 2 so that the `dumpFile()` method returns the average of all the grades input from the file in addition to dumping to the console the contents of the given file. The `dumpFile()` method will now be called as follows:

   `double` average=`dumpFile`(name); // returns average of all grades in a file.

4. Suppose the following declarations are given and are accessible inside the method you are to write.

   int maxStudents=300;// Denotes the max number of students
   String[] name;// Array to save names of students
   double [][]grade;// Array to save grade of each student in the name array
   double [] average; // Array to save averages of each of the four exams

   Write the following methods.

   (a) `readData(String` filename); This method reads data from the file whose name is given as input. Names of students read are saved in the `name` array and grades in the `grade` array. The file contains one line for each student as follows:

   loginID grade1 grade 2 grade 3 grade 4

   where `login ID` is a string without spaces, and `grade1, grade2, grade3`, and `grade4` are numbers in the range 0 to 100 (inclusive).

   (b) `findAverage`(): This method finds the average grade obtained in each exam and saves these in the `average` array. Thus, `average`[0] contains the average on exam 1, `average[`1] contains the average on exam 2, and so on.

   (c) saveAverage(`String` filename): This method saves the averages computed by computed by `findAverage().` The name of the output file is provided as a parameter.

5. You are given the following declarations that are accessible to the method you are about to write.

   JFrame f;
   JButton[] b;
   JTextField t;

   Write a method named `createGUI`() that creates the GUI shown in Figure 1. The GUI has a frame (not a panel) with a 2x2 grid layout.

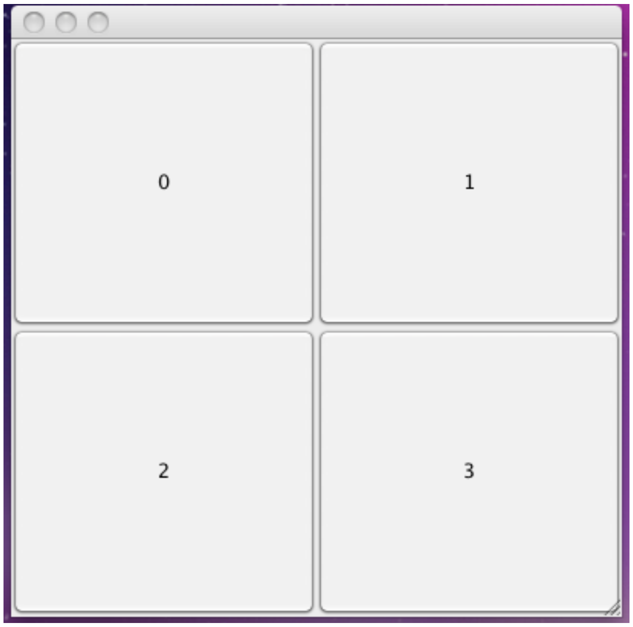
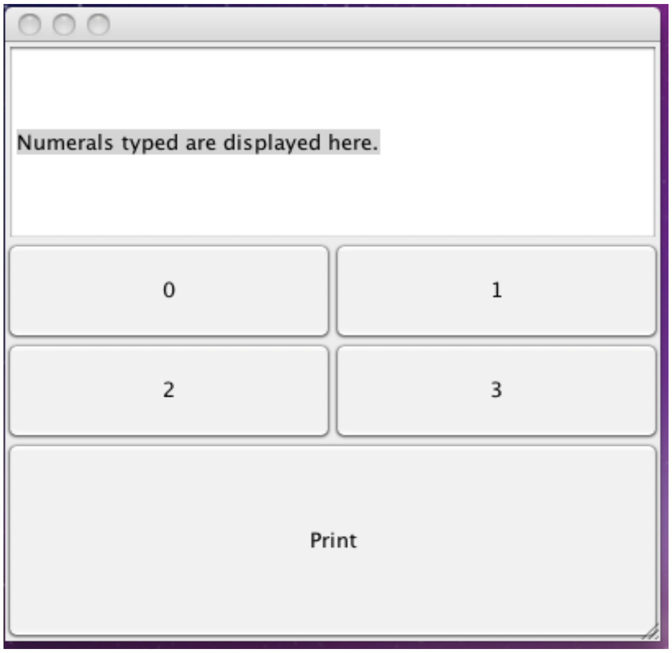


Figure 1

Write *only* the methods needed to perform the following actions: (a) When a button is clicked the number on the button is displayed on the console. (b) When the mouse enters a button, a suitable message, such as "Mouse enters button 2", is displayed on the console. (c) When the mouse exits a button a suitable message, such as "Mouse exited button 3", is displayed on the console.

6. Modify the createGui() method in the previous problem so that the GUI now looks like the one in Figure 2. Assume that the following declarations are already given:

   JFrame f;
   JPanel p;
   JPanel center;
   JButton [] b;
   JButton print**;**
   JTextField t;



Numerals typed are displayed here.

Figure 2

This GUI has a frame on top of which is a panel with a 3x1 grid layout. There is a text field at the top of the panel and a button labeled **`Print`** at the bottom. In the center of the panel there is another panel that has the grid layout and the buttons as in the previous problem.

Write an `actionPerformed`() method that performs the following tasks: On each button click the numeral on the button is displayed in the text field at the top. Thus, if buttons labeled 3, 2, 3, and 1 are clicked in that sequence, the text field will display 3231. When the button labeled `Print` is clicked, the number in the text field is printed on the console and the contents of the text field are reset to the original text, i.e., "Numerals typed are displayed here."

7. Modify the createGui() method in the previous problem so that the method is now called with an integer parameter n>0. The method then creates a GUI with `nxn` buttons labeled from 0 to n2 -1. Thus, if `createGui`() is called as

   `createGui`(3);

   the GUI must have a total of 9 buttons arranged in a 3x3 grid and labeled 0 through 8. The text field and the `Print` button are to be placed just as in Figure 2. The actions performed when a button is clicked are also the same as in the previous problem.

8. Modify the createGui() method in the previous problem so that the method is now called with an integer parameter n>0 as well as a one dimensional array of strings. For example, the method may be called as follows:

createGui(3, label);

where the label array is declared as

String [] label=new String [n*n];

Thus, the label array contains exactly n*n strings. The buttons must be labeled with the strings in the label array such that the leftmost button at the top is labeled with the string in label [0], the next button with the string in label [1], and so on. The last button located at the bottom right must be labeled with the string in label [n*n-1].

9. Modify the createGUI () method in the previous problem so that the layout of the frame (not the panel) is BorderLayout. Add the text field to the north, buttons in the center inside a panel with an nxn grid layout, and the Print button in the south. The west and east do not contain any widget.

<End of Exam 2 Problem Set>