CSCD211 Lab01 - 2-D & Multi-Dimensional Arrays

Instructor: Nancy Ripplinger

Zip Folder Lab Label: CSCD211_Lab01Ch08_firstname_lastname

Send all program files in a zip folder. Label each file as follows

CSCD211_Ch0801_SumArray.java,

CSCD211_Ch0803_SortStudentsArray.java,

CSCD211_Ch0804_WeeklyHoursArray.java,

CSCD211_Ch0816_SortRowsColumnsArray.java

***8.1** Write a method that returns the sum of all the elements in a specified column in a m(*Sum elements column by column*) matrix using the following header:

```
public static double sumColumn(double[][] m, int columnIndex)
```

Write a test program that reads a 3-by-4 matrix and displays the sum of each column. Here is a sample run:

```
Enter a 3-by-4 matrix row by row:
1.5 2 3 4 ↵Enter
5.5 6 7 8 ↵Enter
9.5 1 3 1 ↵Enter
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9.0
Sum of the elements at column 2 is 13.0
Sum of the elements at column 3 is 13.0
```

Listing 8.2 GradeExam.java

```
1  public class GradeExam {
2    /** Main method */
3    public static void main(String[] args) {
4      // Students' answers to the questions
5      char[][] answers = {
6        {'A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
7        {'D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'},
8        {'E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'},
9        {'C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'},
10       {'A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
11       {'B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
12       {'B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
13       {'E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}};
14
15     // Key to the questions
16     char[] keys = {'D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A', 'D'};
17
18     // Grade all answers
19     for (int i = 0; i < answers.length; i++) {
20       // Grade one student
21       int correctCount = 0;
22       for (int j = 0; j < answers[i].length; j++) {
23         if (answers[i][j] == keys[j])
24           correctCount++;
25       }
26
27       System.out.println("Student " + i + "'s correct count is " +
28         correctCount);
29     }
30   }
31 }
```

***8.3 (SORT STUDENTS ON GRADES)** Rewrite **Listing 8.2**, GradeExam.java, to display the students in increasing order of the number of correct answers.

****8.4 (COMPUTE THE WEEKLY HOURS FOR EACH EMPLOYEE)** Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee's seven-day work hours with seven columns. For example, the following array stores the work hours for eight employees. Write a program that displays employees and their total hours in decreasing order of the total hours.

	Su	M	T	W	Th	F	Sa
Employee 0	2	4	3	4	5	8	8
Employee 1	7	3	4	3	3	4	4
Employee 2	3	3	4	3	3	2	2
Employee 3	9	3	4	7	3	4	1
Employee 4	3	5	4	3	6	3	8
Employee 5	3	4	4	6	3	4	4
Employee 6	3	7	4	8	3	8	4
Employee 7	6	3	5	9	2	7	9

***8.16 (SORT TWO-DIMENSIONAL ARRAY)** Write a method to sort a two-dimensional array using the following header:

```
public static void sort(int m[][])
```

The method performs a primary sort on rows, and a secondary sort on columns. For example, the following array

```
{{4, 2},{1, 7},{4, 5},{1, 2},{1, 1},{4, 1}}
```

will be sorted to

```
{{1, 1},{1, 2},{1, 7},{4, 1},{4, 2},{4, 5}}.
```