**CS1180**

**Project 4**

**Purpose:** This program will provide experience using arrays and input and output files.

Write a program that reads student data from an input file named “scores.txt” (provided on the course web site). This file will contain a list of 16 student names and the student’s scores for labs, quizzes, projects, and exams as follows:

Last name, first name List of 15 lab scores List of 12 quiz scores List of 6 project scores List of 2 exam scores Final exam score

Your program is to create the following arrays, with the size of each of these arrays being equal to the number of students in the file (16). The information contained in each array will be read directly from or calculated from the data in the file.

• an array of names (the full name read from the file)

• an array for lab averages (the % of possible points earned for all labs; each lab is worth 10 points)

• an array of quiz averages (the % of possible points earned for all quizzes; each quiz is worth 5 points)

• an array of project averages (the % of possible points earned for all projects; each project is worth 25 points)

• an array of exam averages (the % of possible points earned for all exams; each exam is worth

100 points)

• an array of final exam scores (score given in input file is out of 100 points)

• an array of course averages (calculate based on these weightings: labs, 20%; quizzes, 10%;

projects, 30%; exams, 15%; final exam, 25%)

• an array of course grades (determined on a 90-100 A; less than 90 but at least 80 B, etc. scale)

Once all the arrays are populated, sort the arrays in descending order based on course average. You may use any simple sort method discussed in class (selection sort, insertion sort, or bubble sort). Be sure the parallel correspondence is maintained when sorting.

Output information from all arrays in a neat, columnar format, as shown on the last page of this document.

Display a message to the screen before ending your program stating the name of the output file containing the results.

**General instructions:**

* Use named constants for number of labs, quizzes, etc., and for the number of points that labs are worth, quizzes are worth, etc. Do NOT use literals within your code!
* Your solution should open only one File/Scanner object for input. Pass the Scanner object as a parameter to methods that read from the input file.
* You must use the following methods in your solution:
* A method that displays all information to the output file
* A method that sorts the arrays
* Method(s) that read lab, quiz, project, or exam scores and returns the % earned
* Use other methods where appropriate; all main tasks should be done in a method, one task per method
* Display all numbers with one decimal point
* Be sure to follow the Style Guidelines.
* **Arrays are the ONLY data structure allowed in this program**. You may NOT use ArrayLists, user-defined classes, or any other structure not specifically mentioned in this specification!

**Grading (100 pts total):**

(2) Declares appropriately-named constants for all literal values used in calculations

(3) Declares appropriately-named variables for all input values and all calculated values. Uses appropriate data types

(8) Correctly opens and closes input and output files. Uses a try…catch block, NOT throws clause.

(25) Correctly reads all information from the input file; stores name and final exam score directly; reads values for other categories and calculates/stores the % earned in the appropriate array

(6) Correctly calculates/stores course average and final grade

(10) Correctly sorts the arrays in descending order based on course average

(15) Correctly outputs the information in columns, as shown on the next page

(2) Displays a message to the screen when finished.

(10) Uses methods for all major tasks

(8) Uses proper Javadoc comments above and below the package name and above each method

(5) Uses single-line comments above each major step

(6) Code is properly formatted (i.e. indenting, alignment of braces)

Sample output:

