CS 1050, Mr. Kramer Programming Assignment #10 # points: 120

Purpose Learn how to use two-dimensional arrays. Calculate payroll data using parallel and two‑dimensional arrays.

Due Date Per the Course at a Glance. Cannot be resubmitted.

Submissions In this order: printed copies of the source code with line numbers, the input file and the output file. You can create the input file by copying the data below and pasting it to file YourName\_S\_10\_Input.txt.

Java Topics Two-dimensional arrays, partial array sorting, loops.

References Textbook – use the index for relevant topics. The programming standards and guidelines as discussed in class. Selection Sort in the Toolkit.

Specification Except for the Toolkit and constants note below, do ***NOT*** use static (global) variables. Also, all output to the output file should be echoed on the console, too.

1. Assume that a company has at most 30 employees (static constant). Your program should handle the situation if there are more than 30 rows of data by letting the user know, then terminating the program.

2. Use two parallel arrays that might be partially filled. One is a two-dimensional array -- a row of this array will hold seven values in this order: [0] number of hours worked, [1] hourly pay rate, [2] gross pay, [3] net pay, [4] savings amount, [5] IRA investment amount and [6] taxes deducted.

The second array is a one-dimensional array of Strings holding the names of the employees. The payroll information for the i-th name from the array of Strings will be in the i-th row of the two-dimensional array.

3. This program reads from a file and writes to a file. Data: Each line contains the number of hours worked, the pay rate (both real numbers), followed by the name. One or more spaces separate the fields.

4. Gross pay is calculated as follows:

hours ≤ 40.0 Paid at the pay rate.

40.0 < hours ≤ 50.0 hours > 40.0 and less ≤ 50.0 are paid at one and a half times the rate. The first 40 hours are paid at the pay rate.

50.0 < hours hours > 50.0 are paid at twice the pay rate. The first 40 hours are paid at the pay rate. Hours 40 to 50 are paid at time and a half.

Examples: with a pay rate is $10.00 an hour, gross pay is calculated as follows:

# Hours Gross Pay

30 30 x $10 = $300

40 40 x $10 = $400

46 40 x $10 + ( (46-40) x $10 x 1.5 ) = $400 + (6 x $15) = $490

50 40 x $10 + ( (50-40) x $10 x 1.5 ) = $400 + (10 x $15) = $400 + $150 = $550

58 40 x $10 + ( (50-40) x $10 x 1.5) + ( (58-50) x $10 x 2) = $400 + $150 + $160

= $710

Note how the pay rate ($10 in the example above) can be factored out, which you should do in the program. For example, the reimbursement for 46 hours is

$10 x (40 + (46-40) x 1.5) = $10 x (40 + (6 x 1.5)) = $10 x (40 + 9) = $490

Do you see why?

5. Use a method to calculate the amount that goes towards the IRA investment which is 8% (static constant) of the gross pay. The adjusted gross pay is the gross pay less this IRA investment amount and is not stored in the array of payroll values.

6. The Federal withholding tax rate is 18% (static constant) and the state withholding tax rate is 4.5% (static constant). Use a method to calculate total taxes and round the answer to the nearest penny. Use the ‘roundNumber’ method in Toolkit.

7. Use another method to calculate net pay which equals gross pay minus taxes.

8. Use a method to calculate the total savings amount which is 10% of net pay. The total savings amount is array column 4, the sum of the IRA investment amount and savings.

9. Use a method to create a report -- this method will be called after all input and all calculations are done and will call the three methods specified below.

a. A method to print heading information including:

Fabulous Furniture Company

Payroll Report

and column headings for the name, net pay, gross pay, federal tax, state tax, hours, and pay rate, in that order. The column headings should look like this:

Name Gross Pay Net Pay Savings Taxes Hours Pay Rate

------------------ --------- ------- ------- --------- ----- --------

Note that the IRA investment amount does not appear in the output but is included in the Savings column.

b. A method to print the above detail lines for all employees, matching the headings indicated in part a. Single space the detail lines.

c. A method that calculates the totals for gross pay, net pay, savings, taxes and hours, and prints a summary line with those totals. Under the column with the pay rate, print the average pay rate. (The summary line should print the totals at the bottom of the appropriate columns.) Also, print a line with a message that contains the number of employees processed which could be less than 30.

10. After the report has been printed, sort the employees alphabetically and print out the report again with the same print methods: heading, details and summary. The sort used is located in Toolkit and is called selectionSortStringWithNumbers which is discussed at the end of this document and will be reviewed in class.

11. Next, sort the employees in ascending order by gross pay and print out the header, details and summary again. Use the toolkit’s selectionSortStringWithNumbers for the sort. The methods specified above should be used to print out the report.

12. The main program should consist mostly of method calls. Use one file for all the source code (i.e., no classes). Use appropriate parameter passing and no non-local (static) variables other than the constants specified above.

13. The Data. The input file format is the number of hours (double), the pay rate (double) and a name (which may or may not have an initial). Spaces separate the fields. The file has been uploaded and is named CS1050\_Assignment\_10\_Input.txt. Remember to trim the name once you’ve extracted it from the input file. The String method name is ‘trim’. The data starts on the next page.

41.00 11.74 Hancock John

50.00 10.60 Light Karen L

52.00 10.80 Fagan Bert Todd

62.00 12.24 Antrim Forrest N

38.40 14.74 Camden Warren

44.33 12.04 Mulicka Al B

41.75 13.40 Lee Phoebe

24.00 11.40 Bright Harry

41.00 10.40 Garris Ted

43.00 12.00 Benson Martyne

31.90 12.40 Lloyd Jeanine D

44.00 13.50 Leslie Bennie A

48.40 14.40 Brandt Leslie

42.00 12.90 Schulman David

50.10 10.84 Worthington Dan

70.40 12.66 Hall Gus W

40.10 12.74 Prigeon Dale R

43.00 12.44 Fitzgibbons Rusty

50.00 12.24 Feistner Merle S

23.00 12.34 Hallquist Dottie

43.33 10.90 Bolton Seth

43.00 12.20 Taylor Gregg

42.00 12.94 Raskin Rose

50.10 12.44 Kenwood Pat

44.33 14.64 Slaughter Lew

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Using the Sort Routine in the Toolkit

Here’s how to use the sort routine that’s in the Toolkit. Assume the following declarations:

static Toolkit tools = new Toolkit(); // Access the Toolkit

static final int MAX\_NAMES = 30; // Maximum number of names allowed

// Here are the two arrays containing employee names and

// corresponding values. Column 2, as defined above, is the gross

// pay column. nNames contains the number of names read.

String[] names = new String[MAX\_NAMES];

double[][] values = new double[MAX\_NAMES][6];

int nNames = 0; // Number of names read into the names array

…

…

// Step 10 calls for sorting alphabetically. Here’s how:

tools.selectionSortStringWithNumbers(names, values, nNames, -1);

// Step 11 calls for sorting by gross pay, column 2. Here’s how:

tools.selectionSortStringWithNumbers(names, values, nNames, 2);