CS 1050, Mr. Kramer Programming Project #3 Points: 60

Purpose Calculate mileage reimbursements using arrays and methods.

Due Date As discussed. Can be resubmitted once, must work in teams.

Submissions In this order: printed copies of the source code with line numbers, the input file, and the output file. As always, write your name, section and project number in the upper right corner.

Java Topics Arrays, Java I/O, while loops, for loops, if-else-if, methods, summing/averaging data.

Use of leftPad in the ToolkitBasic class.

References Textbook – use the index for relevant topics

The programming standards and guidelines as discussed in class

ToolkitBasic.java

Specification

The Mathematical Association of America hosts an annual summer meeting. Each state sends one official delegate to the section officers’ meeting at this summer session. The national organization reimburses the official state delegates according to the scale below. Write a Java program to calculate the reimbursement values, satisfying the specifications below. Details on array and method usage follow these specs.

1. The main method should declare all key variables at the start of the method.
2. Read data from a file and write data to a file. **All data written to the output file should be echoed on the console.** There should be only one input file and only one (separate) output file. The files are named, respectively,
   1. YourName\_S\_03\_Input.txt YourName is FirstnameLastname
   2. YourName\_S\_03\_Output.txt “S” is the section number
3. The first line of the input file contains an integer number of data values to process. After the first line of the input file, each line contains a type double number which is the number of miles.
4. Use an “if/else/if” construct and the scale below to calculate the mileage reimbursement if the input value is > 0.
5. Output the results to the file and to the console in a table format. (See below for a sample layout.) There should be a heading for each of the table’s columns. Print one line of output for each mileage value processed. The columns of the table should be lined up by the decimal point using the leftPad method. Each detail line of the table will contain the number of miles (double – print with one decimal place) and the reimbursement amount (double – print with two decimal places). If the input value is <= 0, output five stars in place of the reimbursement amount.

Using a text editor or jGrasp’s File > New > Plain Text, you will need to create an input file containing the date below. Use this data in the input file to your program. **DO NOT change these numbers.** The first number is the number of numbers to process.

10 250.6 99.4 -2.78 999.4 799.4 1899.8 0 1300.2 1101.7 3333.3

Reimbursement scale:

Round trip mileage Rate

less than 400 miles 18 cents per mile

≥ 400, < 900 miles $ 65.00 plus 15 cents for each mile over 400

≥ 900 miles, < 1300 miles $115.00 plus 12 cents for each mile over 900

≥ 1300, < 1900 miles $140.00 plus 10 cents for each mile over 1300

≥ 1900, < 2600 miles $165.00 plus 8 cents for each mile over 1900

≥ 2600 miles $195.00 plus 6 cents for each mile over 2600

Be sure to appropriately document this program as has been discussed in class and done in previous projects.

Discussion

1. Outlining the program

2. Methods to use

3. How to test the program (sample input file: 1 250.6. i.e., use one number at a time in each scale range)

4. The leftPad method (time permitting: develop the method in class)

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

Sample Layout for the Report

Mileage Reimbursement

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

250.6 (calc’d reimbursement)

99.4 (calc’d reimbursement)

-2.8 \*\*\*\*\*

etc.

Follow the detail lines with the summary information as discussed below.

You will use two parallel one-dimensional arrays of length n, where n is the number of mileage values to process, one to hold mileage values and one to hold reimbursement values. Note that n is determined by reading the first entry in the input file which has the number of entries that follow.

The main purpose of this project is to practice using arrays in methods. Therefore, no arrays or related counts and averages should be declared globally (static before the main program begins). You can declare constants like tab and new line as global variables if you want.

Your program must satisfy the requirements shown below. The main program should be mostly (but not necessarily only) method calls with appropriate parameters.

1. A method that explains the program to the user.

2. A method with the mileage array as a parameter that reads the mileage values and stores them in the mileage array. Use the same approach as in Project 1 to read the file. This method doesn’t calculate the reimbursement; it only reads the data. From this method, determine the number of elements in the possibly partially-filled array and return that number.

3. A method with the two arrays and the number read as parameters that calculates the array of reimbursement amounts. If the mileage value is negative or 0, the corresponding reimbursement amount should be zero.

4. One method prints the heading and another method prints the detail lines in the table. The latter method will have the two arrays and the number read as parameters.

Each detail line of the table contains the number of miles (type double, print with one decimal place) and the reimbursement amount (type double, print with two decimal places). If the input value is <= 0, output five stars instead of the reimbursement amount. Use the leftPad method to format the numbers to one or two decimal places as needed and line up the decimals in the columns.

5. After the array of reimbursements has been calculated, a method or methods calculate the average reimbursement and the average number of miles traveled. These averages will be the averages for mileage values which are > 0. In other words, you might divide by a number less than the number of elements (which, by the way, could be 0, so check for it before you calculate the average).

6. A method outputs the summary information at the end of the table. This output should include the total of the reimbursement values, the number of mileage values processed, and the number of mileage values that were > 0. In addition, output the total of the mileage values that were > 0, and the two averages calculated in the previous step, all with appropriate messages.