GEORGE MASON UNIVERSITY | IT 106
Introduction to IT Problem Solving Using Computer Programming

Programming Assignment 9/10

Scenario:

Based on your new found problem solving expertise, you have been hired at Lisa's Lemons, a cutthroat used car dealership. Lisa is very mean and expects perfection. She wants you to help create a program that helps identify which of her salespeople are doing an excellent job and which of her people are not. She has asked you to determine this using last month's sales based on the following rules:

- An employee is considered to be doing an excellent job if their sales are MORE than the average sales of all salespeople, based on last month's sales.
    - Each of these individuals will earn a 5% bonus based on their last month's sales
- An employee is considered to not be doing an excellent job if their sales are LESS than the average sales of all salespeople, based on last month's sales.
    - Each of these individuals should be reprimanded they are not doing an excellent job and should try harder.
- An employee is considered to be doing an OK job if their sales are EQUIVALENT to the average sales of all salespeople, based on last month's sales.
    - Each of these individuals should be told they are just doing average work

Create a ***modular*** program that will take in the names of each salesperson and the amount of their sales for the last month. Sales must be between $0 and $500,000 (inclusive), or the user must be informed of an error and re-prompted. The program will determine if the employee should earn a bonus (and its amount), a reprimand for not having sales more than the average sales of all salespeople, or nothing at all for having average sales. Finally, the program will display a well-formatted report containing a list of all salesperson names, their amount of sales for the last month, and a note of the bonus amount they have earned, a note to warn the salesperson to increase their sales, or a note to let the salesperson know they are just doing average work.

Other Requirements:

- You must check that all numerical input is valid
- Your solution must use one or more arrays. You may not use any ArrayLists.
- The program must allow up to 10 salespeople to be entered, though there may be fewer. The program must gracefully handle the entry of an 11th salesperson, informing the user that the program only supports 10 salespeople)

To Do (Check Blackboard for Due Dates):

Programming Assignment 9: Solution Design

1) Create a defining diagram that shows the input, processing, and output
2) Create a structure diagram/hierarchy chart grouping processes from the defining diagram into ***modules*** (You need to do this, but you do not need to turn it in as part of the grade)
3) Create a ***modular*** solution algorithm using pseudocode
4) Show testing using the desk checking table method, to include test data, expected results, and a desk checking table. Make sure your desk checking considers multiple cases including both valid and invalid test data to prove your algorithm will work

Upload a Word document containing only items above to Blackboard.

Grading Criteria	
Requirement	**Points**
Defining Diagram with input, processing, and output	40
Efficient Solution Algorithm	40
Thorough Desk Checking Table including test data, and expected results	20

Full points will be awarded for an accurate, efficient, complete defining diagram, solution algorithm, and desk checking table. Partial credit is available.

Programming Assignment 10: Solution Implementation

Write a well-documented, efficient Java program that implements the algorithm you identified. Include appropriate documentation as identified in the documentation expectations document.

Note: You may not use the Scanner or System.out classes. For input/output, you must use the JOptionPane class.

Upload the .java file of the final program to Blackboard.

Full points will be awarded for an accurate, efficient, complete Java program. Partial credit is available. Any final program that does not compile will receive an automatic zero.

Grading Criteria	
Requirement	**Points**
Implementation of Java Program, using efficient practices where appropriate, such as the use of constants, good variable names, no redundant code, etc.	70
Appropriate objective-style documentation	10
Appropriate intermediate comments	20