Assignment 2 - 100 points

(Note: Exam 1 assumes you’ve finished this asgn)

Objectives – to practice these new concepts:
• decision-making with if, if-else, if-else-if-… switch
• data validation for user input using a while (or while/do) loop
• nicely formatted output report with printf and format strings
• multiple input data sets using a while (or while/do) loop
• named constants

PROJECT OVERVIEW
This project provides a detailed payroll report for the sales staff at TheLaptopShop. The company has 3 tiers of salespeople: low, middle, high (designated by L, M, H) – based on their past sales record. Each month they might get promoted to the next higher tier (or, if at the highest level already, receive a super bonus instead). The store sells 3 different laptop models: basic for $450.90, mid-range for $850.50 and high-end for $1350.95

A person’s pay includes 4 components:
1. base salary
2. commission - based on the number and types of laptops he/she sold that month:
   - $50 for each basic laptop, $100 for each mid-range laptop, $150 for each high-end laptop
3. bonus - based on categories for his/her total sales that month (see below)
4. super bonus – based on very high sales, if they’re already in the highest tier (see below)

Bonus is based on a person’s total sales amount and is calculated as follows:

<table>
<thead>
<tr>
<th>Total sales categories</th>
<th>Bonus amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to $2,500</td>
<td>0% of total</td>
</tr>
<tr>
<td>$2,500.01 - $5,500</td>
<td>1% of total</td>
</tr>
<tr>
<td>$5,500.01 - $10,500</td>
<td>$75 + 2% of (total minus $5,500)</td>
</tr>
<tr>
<td>$10,500.01 - $13,500</td>
<td>$125 + 3% of (total minus $10,500)</td>
</tr>
<tr>
<td>Above $13,500</td>
<td>$375</td>
</tr>
</tbody>
</table>

Tier Promotion & Super Bonus:
If a salesperson earns a commission which is 75% or more of their base salary, then they’re EITHER:
1) promoted to the next higher tier (L → M or M → H)
2) OR, if they’re ALREADY in the highest tier (H), the person earns a SUPER_BONUS instead
   (a named constant currently set to $1000) which is added to their monthly pay.

STAGES of the PROGRAM DEVELOPMENT
0) program hardcodes the input data for ONE person, does the appropriate calculations and produces the output payroll report for that person (see below for exact format)
1) AFTER STAGE 0 WORKS CORRECTLY
   program gathers input data for one person from the user instead of using hardcoded data
2) AFTER STAGE 1 WORKS CORRECTLY
   add data validation (see rules below) for the input gathering
3) AFTER STAGE 2 WORKS CORRECTLY
   add looping so that the program can handle as many salespeople as the user provides.
   (After handling one person, the program asks whether the user wants to enter data for another person, then proceeds accordingly).

Stage 3 version of the program is the one that you should submit to get the full 100 points.
INPUT (for 1 person)
Use dialog boxes with appropriate prompts to get the following 6 pieces of information from the user:
name, base salary, tier (L for low, M for middle, H for high),
number of basic laptops sold, number of mid-range laptops sold, number of high-end laptops sold

INPUT VALIDATION (added in stage 2)
- Tier values must be one of: L, M, H (or l, m, h)
  - If it’s invalid, use the default, L, and tell the user what happened.
- Base salary must be a positive number (integer or real number) that’s <= MAX_VALID_SALARY (a named constant set to $10,000 for now).
  - If it’s invalid, keep asking the user for a valid salary amount until they enter something correct.
- Number of basic, mid-range and high-end laptops must each be a positive integer that is <= MAX_VALID_UNITS (a named constant set to 50 for now).
  - If it’s invalid, keep asking the user for a valid number of units until they enter something correct.
  - Each of the 3 types of laptops will have its own validation loop.

INPUT - MULTIPLE SALESPEOPLE (added in stage 3)
After printing the report for a single person, ask the user if they wish to enter another salesperson (YES or NO).
If they say YES (or Yes or yes or YES or yeS or …) or Y (or y), then go around the big loop again.
(Use the String method toUpperCase() to reduce the number of cases to just YES and Y)
Any other response (like NO or nope or naw or whatever or who cares or …) will be assumed to be a NO.

PROCESSING (for 1 person)
Compute their monthly pay (base salary + commission + bonus + perhaps the SUPER_BONUS too).
Determine if they moved up a tier or not.

OUTPUT to the CONSOLE (a sample for 1 person)
[NOTE: the data below aren’t necessarily correct – I’m just showing you the FORMAT to use]

Salesperson:  MARIA GONZALES
Starting Tier: High
New Tier:      High
Laptops Sold:  9 basic, 15 mid-range, 12 high-end
Base Salary:   $ 800.20
Commission:    $ 1,200.00
Bonus:         $ 152.28
Super Bonus:   $ 1,000.00
---------------
Monthly Pay:   $ 3,152.28

Congrads, you got the SUPER BONUS since you were already at the High tier

-------------------------------
[alternative message if they got promoted (though no super bonus)]
Congrads, you got promoted to the Middle tier
[alternative message if they didn’t sell >= 75% of their base salary]
Sorry, you didn’t get promoted to the next tier this month

OTHER PROGRAMMING REQUIREMENTS
- follow JavaNamingConventions (shown on course website, 2nd lecture) for variables, constants, project name
- use printf’s, mainly, for the output report