CS 2336 PROJECT 2 – Disneyland Dining Rewards (Standalone Project 1)

Pseudocode Due: 2/8 by 11:59 PM

Project Due: 2/22 by 11:59 PM

Submission and Grading:

- All project deliverables are to be submitted in eLearning.
- The pseudocode should be submitted as a Word or PDF document and is not accepted late.
- Zip all of the source files into a single zip file for submission.
- Projects submitted after the due date are subject to the late penalties described in the syllabus.
- Programs must compile and run with JDK 8.
- Each submitted program will be graded with the rubric provided in eLearning as well as a set of test cases. These test cases will be posted in eLearning after the due date. Each student is responsible for developing sample test cases to ensure the program works as expected.
- **Type your name and netID in the comments at the top of all files submitted.**

Objectives: Create base classes and derive new classes using inheritance. Utilize multiple classes in the same program.

Problem: Disneyland has begun a service that allows guests at the park to pre-order drinks and pick them up at a designated time so as to avoid long concession lines. Guests that spend more than $150 (cumulatively) become preferred customers and are awarded discounts on future orders. A program is needed to track the amount spent by each customer and promote them to preferred customer when they have accumulated $150 in purchases.

Classes:

- **Customer (base class)**
  - Members
    - First name
    - Last name
    - Guest ID
    - Amount spent
  - Methods
    - Overloaded Constructor
    - Accessors
    - Mutators
- **Preferred Customer (derived class)**
  - Member
    - Discount Percentage
  - Methods
    - Overloaded Constructor (chained w/ base class constructors)
    - Accessor
    - Mutator

Details:

- The price of the drink is determined by the type of drink and the amount of ounces ordered
  - The number of ounces ordered and the price per ounce will be given for each order
- Each drink will come in a cylindrical container
- The container can be personalized with different Disney graphics
- The total price of the personalization is determined by the price per square inch of the container
  - The radius, height and price per square inch will be given for each input
- Preferred customer information and regular customer information will be stored in files and will be read into memory before the program processes orders.
  - Note that the files may be empty or may not exist.
  - In both cases, the respective customer array should not be created until a customer reaches preferred status.
- Regular customers and preferred customers will be held in 2 separate arrays.
- The arrays are to be no larger than the actual number of members in each category.
- Orders will be read from a file.
- After processing an order, if a customer has accumulated $150 or more, promote the customer to preferred status.
  - The preferred customer array will be resized to add one more element.
  - Add the new preferred customer into the open element at the end of the preferred customer array.
  - Resize the regular customer array and remove the customer data that was promoted to preferred status.
- Preferred customers get a discount based on how much they have spent overall
  - $150 = 5% discount
  - $200 = 7% discount
  - $350 = 10% discount

Input: Input data will be stored in three files: preferred.dat, customers.dat and orders.dat. The total number of lines in each file will be unknown.

Preferred.dat will hold the data for known preferred customers and customers.dat will hold the data for known regular customers. Both files should be read at the beginning of the program to establish the preferred customer and regular customer arrays respectively. The files may be empty or may not exist. In such cases, there should not be a respective customer array created before reading the orders.

Each line of preferred.dat will be formatted as follows (except for the last line of the file which will not have a newline character). The data will be listed in the following order on each line with a space between each field. There will be no invalid data on any line.

- `customer ID`
- `first name`
- `last name`
- `amount spent`

- `discount`
- `newline`

Each line of customer.dat will be formatted as follows (except for the last line of the file which will not have a newline character). The data will be listed in the following order on each line with a space between each field. There will be no invalid data on any line.

- `customer ID`
- `first name`
- `last name`
- `amount spent`
- `newline`

Orders.dat will hold all orders to process. Each line of the file will be formatted as follows (except for the last line of the file which will not have a newline character). The data will be listed in the following order on each line with a space between each field. There will be no invalid data on any line.

- `customer ID`
- `container radius`
- `container height`
- `ounces`
- `ounce price`
- `square inch price`
- `quantity`
- `newline`

For each order, the main thing to note is the total cost and its connection to the appropriate customer. The total cost will be added to the amount spent for each customer. If the customer has preferred status, the discounted price will be added to the amount spent. After the amount spent has been updated, check to see if the customer has gained preferred status or if the preferred customer qualifies for a larger discount.

When moving a regular customer to the preferred customer array, always add the data to the end of the array. When removing the customer from the regular customer array, shift the elements toward the front of the array. Do not rearrange the order of the data in either array. Remember that each array should be no bigger than the number if elements it needs to hold.

Output: At the end of the program, write the regular customer data and preferred customer data to the respective files. Use the proper format as listed above in the Input section. There will be no output to the screen.