CS 116 Final Project Fall 2019

One submission per lab group.
Design due in Blackboard Tuesday 11/12/19, by midnight
Code (and resubmission of design for re-grading) due in Blackboard Wednesday, 12/04/19 by midnight

Objectives:

1. (20 points) Design a multi-object application.
2. (80 points) Code a multi-object application.

Office Supply Order System - A supply order consists of a unique order ID (numeric), a customer ID(alpha), a product ID(alphanumeric), a date of the order, an amount, and if the order is a "one-time" or "repeated" order. For "repeated" orders, a supply order also has a period (the number of days between repeats), and an end date for the repeating. The Office Supply Order System needs to allow a Order Manager to manage a set of supply orders: to be able to add an order (and assign a unique order ID), delete an order (given the order ID), and list all the orders for a particular customer ID in increasing date sorted order. Finally, the Office Supply Order System needs to be able to calculate and report for each month on the inventory needed for each product ID based on the current supply orders.

Each class for the Office Supply Order System needs its own individual test program. Each class for the Office Supply Order System needs its own individual exception class and should throw exceptions for any invalid action. Other classes need to catch these exceptions where appropriate.

Create a UserInterface class for your Office Supply Order System. This should be the project "void main" that allows the user these options:

1. specify a file name of orders to load
2. specify the information to add an order or repeated order
3. specify an orderID to delete
4. specify a CustomerID to display a list of orders for that customer in increasing date sorted order
5. calculate and output an order inventory report (sorted by year, month and productID).

Sample input file (orderssmall.txt) for the project can be found in Blackboard in Final Project folder It is a comma delimited file with following fields: O or R (onetime or repeated order), customerID, productID, orderDate, orderAmount, and if a repeated order, period and endDate.

You will need to write an input file generator that will generate random input files. This should be its own "void main". Assume:

- customerIDs comes from the current NYSE Company list (see companylist.txt file in Final Project folder in Blackboard)
- productIDs are all a single captial letter followed by a single numeric digit
- orderDates are all in 2019
- orderAmounts are all integers less than 100
- repeatedOrderPeriods are all 1 to 30 days
- repeatedOrderEndDates all in 2019

Design Requirements - Create the following for the above requirements (see the UML Design Reference document in Final Project folder in Blackboard for help):

1. (2 points) Create a list of nouns and noun phrases from the above Requirements Document.
2. (2 points) Create Class Diagrams only for the nouns and noun phrases that have significance.
3. (3 points) Show the associations between the classes and any classes that are compositions of other classes.
4. (5 points) Add Attributes to the Class Diagrams created earlier.
5. (5 points) Add Operations to the Class Diagrams created earlier.
6. (3 points) Create test cases for each class.

Coding Requirements

1. (80 points) Code and test your Office Supply Order System