Homework #2 Inventory processing using Stacks and Queues

Due: 6/4/2024

Write a program to handle the flow of inventory into and out of a warehouse. The warehouse will have numerous deliveries of new inventory and orders for inventory. The inventory in a filled order is billed at a profit of 50 percent over their cost. Each delivery of new inventory may have a different cost associated with it. The accountants for the firm have instituted a last-in, first-out system for filling orders. This means that the newest items in inventory are the first ones sent out to fill an order. Also, the most recent orders are filled first. This function of inventory can be represented using two stacks: orders-to-be-filled and inventory-on-hand. When delivery of new inventory is received, any unfilled orders (on the orders-to-be-filled stack) are processed and filled. After all, orders are filled. If there is inventory remaining in the new delivery, a new element is pushed onto the inventory-on- hand stack. When an order for new inventory is received, one or more objects are popped from the inventory-on-hand stack until the order has been filled. If the order is completely filled and there is inventory left over in the last object popped, a modified object with the quantity updated is pushed onto the inventory-on-hand stack. If the order is not completely filled, it is pushed onto the orders-to-be-filled stack with an updated quantity of inventory to be sent out later. If an order is completely filled, it is not pushed onto the stack.

Write a class with functions to process the shipments received and to process orders. After an order is filled, display the quantity sent out and the total cost for all the inventory in the order. Also, indicate whether there is any inventory remaining to be sent out at a later time. After delivery is processed, display information about each order that was filled with this delivery and indicate how much inventory, if any, was stored in the object pushed onto the inventory-on-hand stack.

**Running examples:**

Process order 2

Finished processing new order

Process inventory 5

Processed 2 items from most recent order;

Order processed completely and removed from stack.

Cost:5

Selling price: 7.5

total order: 15

3 items in shipment;

shipment pushed into stack

items cost $5

Finished processing new Inventory

Process inventory 2

2 items in shipment;

shipment pushed into stack

items cost $10

Finished processing new Inventory

Process inventory 2

2 items in shipment;

shipment pushed into stack

items cost $7

Finished processing new Inventory

Process inventory 1

1 items in shipment;

shipment pushed into stack

items cost $15

Finished processing new Inventory

Process order 10

Processed 1 items for order, obtained from most recent shipment;

no more items in shipment, removed from stack.

Cost: 15 Selling price: 22.5

total order:22.5

Processed 2 items for order, obtained from most recent shipment;

no more items in shipment, removed from stack.

Cost: 7 Selling price: 10.5

total order:43.5

Processed 2 items for order, obtained from most recent shipment;

no more items in shipment, removed from stack.

Cost: 10 Selling price: 15

total order:73.5

Processed 3 items for order, obtained from most recent shipment;

no more items in shipment, removed from stack.

Cost: 5 Selling price: 7.5

total order:96

Finished processing new order

Please use the attached header files for this assignment. Don’t make any changes to the names of the files or functions. You just need to implement ProcessOrders.cpp based on the three header files that you are given for this assignment. You also must submit a cxxtest program similar to the running examples to test your program. Use the examples to verify that your program works. But use different quantities and cost values for the shipments or orders in your cxxtest program when you submit the assignment.