Assignment 4 – CSC 229 Data Structures and Algorithms I

Due: 11/16/2022 @ 10:50am

Submission Guidelines

Create a winzip file containing the WHOLE project directory and submit on Blackboard.

IMPORTANT – Make sure you ***properly comment*** AND ***properly indent*** your program. The commenting and indenting documents are on Blackboard. ***If you fail to properly comment or properly indent points will be deducted.***

VERY IMPORTANT – IF THE PROGRAM DOES NOT COMPILE THERE WILL BE MAJOR POINTS TAKEN OFF. THIS MEANS IT WILL BE A FAILING GRADE.

Overview

You will have to implement the following:

- Write a linked stack of Purchase.
- Write a linked queue of Purchase.

Suggestion: You are not required to but it is strongly advised that you write code in main to make sure that each method works as it should.

Part 1 – Purchase Class

Use the Purchase class from assignment 1.

Part 2 – Create a PurchaseStack Class

Write a linked stack. It should store instances of Purchase.

PurchaseStack Class Specifications

1. Create private members as necessary.

2. Your class must implement the following methods (use the given method headers):
    a. Default constructor.
    b. Copy constructor (should be a DEEP COPY of the parameter). Here is the prototype: PurchaseStack(PurchaseStack other)
    c. void push(Purchase g) – Adds the Purchase instance to the front of the list.
    d. void pop() – Removes the top element from the stack (does not return anything).
    e. Purchase top() – Returns the purchase at the top of the stack.
    f. void makeEmpty() – Clears the stack.
    g. int getLength() – Returns the number of purchases being stored in the stack.
    h. PurchaseStack makeCopy() - Write a makeCopy method (should be a DEEP COPY of the current instance).
    i. Implement an equals override. It should return a value based on the "values" and not the "references". It should return true if all purchase data stored in the stack has the same values and is in the same order.

Part 2 – Create a PurchaseQueue Class

Write a linked queue. It should store instances of Purchase.

PurchaseQueue Class Specifications

1. Create front and rear member variables.
2. Your class must implement the following methods (use the given method headers):
    a. Default constructor.
    b. Copy constructor (should be a DEEP COPY of the parameter). Here is the prototype:
    PurchaseQueue(PurchaseQueue other)
    c. void enqueue(Purchase g) – Adds the Purchase instance to the end of the queue.
    d. Purchase dequeue() – Removes and returns the front element of the queue.
    e. void makeEmpty() – Clears the queue.
    f. int getLength() – Returns the number of purchases being stored in the queue.
    g. PurchaseQueue makeCopy() - Write a makeCopy method (should be a DEEP COPY of the current instance).
    h. Implement an equals override. It should return a value based on the "values" and not the "references". It should return true if all purchase data stored in the queue has the same values and is in the same order.