**Lab 11:** LIFO vs FIFO Linked Lists

EGR/CSC1054

100 points total

Part I (30 points) – Creating a Stack of Cards with LIFO Linked Lists

Create a Card and Node classes (10)

* Card Class
  + It has a single member, int num.
  + It should have a toString that prints “Card: <integer>”.
* **Node Class: The Card class should not be the node class. The Node’s only data should be the Card.**

Create the Linked List Stack class (10)

* Create a linked list **stack** class for cards. Build off the linked list stack code from class.
* You must have push, pop, peek, and toString in your stack class.
  + toString should print out the entire stack indicating the top, what cards are present, and the bottom. (see output examples below).

Create the client (10)

* Create a new Stack.
* Add the numbers 1, 2, …, 6 to the stack.
* Print out the stack.
* Pop the stack 3 times.
* Print **only** the top of the deck.
* Add the numbers 7, 8, and 9 to the stack.
* Print out the stack again.

Example Run

Stack:

<top>

Card: 6

Card: 5

Card: 4

Card: 3

Card: 2

Card: 1

<bottom>

Card: 3

Stack:

<top>

Card: 9

Card: 8

Card: 7

Card: 3

Card: 2

Card: 1

<bottom>

Part II (70 points) – FIFO Linked List at a Grocery Store

(10 points) Create a Customer and Node classes

* Customer
  + It has a name and number of items.
  + It also has any necessary methods and a toString that prints:

“Name: <name> / Items: <items>”

* **Node Class: The Customer class should not be the node class. The Node’s only data should be the customer**

(45 points) Create a Checkout Line class

* **Treat this class as if it was your LinkedList Queue class** – however, instead of containing variables for only one list, this will have variables for two different lists inside:
  + One for the normal line (customers with more than 15 items)
  + One for the fast-track line (customers with 15 items or less)
  + It should not simply contain two queues.
* Use any other variables needed to make these next methods work
* It should have methods for enqueue, dequeue, peek, isEmpty, and toString
  + (10 points) enqueue - when called, it should determine which line the customer should go into based on the number of items.
  + (10 points) dequeue when called, it should go back and forth between one person in the normal lane and one in the fast lane regardless of the order added.
    - **Depending on whichever line gains a node first, that line should be called first.**
      * **Suggestion: have an integer where 0 indicates both queues are empty, 1 indicates fast next, and 2 indicates normal next.**
    - If one of the lines are empty, but the other one is not, then you should print from the other line
    - (Example: The normal line should be dequeued next, but it is empty. However, there are people in the fast line, so you will dequeue from the fast line.)
  + (10 points) peek - when called, it should return the next person that would be removed with dequeue
  + (5 points) isEmpty – it should check if both lines are empty
  + (10 points) toString – when called, it will print out the complete line (look at example below)
    - the toString should go back and forth between the two lines like dequeue/peek
    - **Hint**: use temporary variables to go through the line and not lose the information from the original copy!
  + **Your program should not crash.** Dequeue, Peek, and toString should indicate if there is no one in the lines anymore.

(15 points) Create the client

* Create a Checkout Line of Customers.
* Create a menu/loop to enable the user to
  + line up customers
  + have the next customer check out
  + see how long the line is
  + quit
* You may assume that the user will input the information correctly

Example Run

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

There is no one in line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

No one is in line to check out!

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 1

Enter the customer’s name and number of items:

>> Chris 17

Chris with 17 item(s) is in the normal line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 1

Enter the customer’s name and number of items:

>> Irene 20

Irene with 20 item(s) is in the normal line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

Line:

Name: Chris / Items: 17

Name: Irene / Items: 20

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 1

Enter the customer’s name and number of items:

>> Felix 12

Felix with 12 item(s) is in the fast-track line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

Line:

Name: Chris / Items: 17

Name: Felix / Items: 12

Name: Irene / Items: 20

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

Chris is checking out with 17 item(s).

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

Line:

Name: Felix / Items: 12

Name: Irene / Items: 20

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 1

Enter the customer’s name and number of items:

>> Kevin 8

Kevin with 8 item(s) is in the fast-track line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

Line:

Name: Felix / Items: 12

Name: Irene / Items: 20

Name: Kevin / Items: 8

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

Felix is checking out with 12 item(s).

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

Irene is checking out with 20 item(s).

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

Kevin is checking out with 8 item(s).

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 2

No one is in line to check out!

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 3

There is no one in line.

What would you like to do with the supermarket line?

1 – have a customer line up

2 – have a customer check out

3 – check who is in line

4 – quit

>> 4

Thank you for shopping with us!