**Assignment 4 – Linked List Implementation**

*Much easier to read on Microsoft Word if you:*

1. *Go to Review > Editor*
2. *Click on the “Total Suggestions” button*
3. *Spam click one of these buttons (your choice) when they show up:*
   1. *“Ignore once” or “Don’t check for this issue”* ***and***
   2. *“Ignore all” or “Add to Dictionary”*

The purpose of this assignment is to get familiar with the linked list. Create a project called *Assignment 4*. You will then write four classes:

**Assignment4.java** – A driver class that contains only a main method which should:

Create a new *CustomerList* object

Prompt the user to enter a file name

Call the *getCustomerList()* method with the file name that the user entered

while (true) {

// Prompt user to enter command

if (command is “a”) {

// Prompt user for customerNumber firstName lastName balance

// Create a customerRecord and insert the customer record at the head of

// a singly linked list.

} else if (command is “f”) {

// Prompt the user to enter a customer number and then display the

// corresponding record on the screen.

} else if (command is “d”) {

// Delete the first customer record from the singly linked list.

} else if (command is “r”) {

// Prompt the user to enter a customer number.

// Delete the given customer record corresponding to the number from the

// singly linked list.

// If the customer number does not exist, please show the error message.

} else if (command is “n”) {

// Call *getTotalCustomerNumber()* method and then display the total number

// of customer’s in the database on the screen.

} else if (command is “q”) {

// Prompt the user to enter a file name to save the latest update.

// Save the information in the database to the file specified by user.

// Terminate program.

} else {

// Display error message.

}

}

**customerList.java** – This class contains **a singly linked list** of customerRecord objects. It should have the following data attributes and methods:

public class customerList {

**private Node head** – This is the reference variable to the first node of the linked list

**public customerList()** – Build the list object

**public void getCustomerList(String fileName)** – Reads a text file and fills the ***linked list*** with the records from the file

**public customerRecord getCustomerRecord(int customerNumber)** – Returns the object corresponding to the customer with the specified customer number. If the customer number is not in the linked list, return *null*. (check search function)

**public void enterCustomerRecord(customerRecord new\_record)** – Inserts the customer record at the head of a singly linked list. (check insert function)

**public void removeFirstCustomerRecord()** – Delete the first customer record from the linked list. (check removeFirstNode)

**public int getTotalCustomerNumber()** – Counts the number of customer’s in a singly linked list

**public void saveCustomerList(String filename)** – Save the information stored in the linked list to the file called *filename*

**public void removeCustomerRecord(int customerNumber)** – Delete the given customer record corresponding to the customer number from the linked list. If the customer number does not exist, please show the error message. (check removeNode)

}

/\*\* Node of a singly linked list of customerRecord \*/

public class Node {

private customerRecord element;

private Node next;

/\*\* Creates a node with the given element and next node \*/

public Node(customerRecord s) { element = s; next = null; }

/\*\* Returns the element of this node. \*/

public customerRecord getElement() { return element; }

/\*\* Returns the next node of this node. \*/

public Node getNext() { return next; }

// Modifier methods:

/\*\* Sets the elements of this node. \*/

public void setElement(customerRecord newElem) { element = newElem; }

/\*\* Sets the next node of this node. \*/

public void setNext(Node newNext) { next = newNext; }

}

**Node.java**

**customerRecord.java** – A class that contains the following data attributes and methods:

**private int customerNumber** – A unique number assigned to each customer

**private String firstName** – The customer’s first name

**private String lastName** – The customer’s last name

**private float balance** – The customer’s balance

**get/set Methods** for each data attribute

**public String toString()** – Special method to be used when printing a *customerRecord* object

**Comments and Documentation**

*(****-30%*** *of grade if none)*

You need to add appropriate comments and documentation for methods *and variables* in these programs.

**Trevor’s Comments and Documentation Style for Methods *(For Copy/Paste)***

//!=====< NAME >=====

// BEHAVIOR:

//

// PARAMETERS:

//

// RETURNS:

//@============================================================

*(If not using “Better Comments” extension on Visual Studio Code, replace ‘!’ and ‘@’ with* ***a space*** *or another ‘****=’****.)*

**Dr. Zhong’s Comments and Documentation Style for Methods *(For Copy/Paste)*:**

///////////////////////////////////////////////////////////

// NAME:

// BEHAVIOR:

//

// PARAMETERS:

//

// RETURNS:

///////////////////////////////////////////////////////////