```java
// Rachel Friedman | February 27, 2020 | Data Structures 3130 | Assignment 3 | Main.java

import java.text.NumberFormat;
import java.util.*;
import java.io.File;
import java.io.FileNotFoundException;

public class Main {
    static double discount = 0;
    static boolean promotion = false;
    static int count = 0;
    LinkedList<Widget> widgets = new LinkedList<Widget>();

    public static void main(String[] args) {
        Main main = new Main();
        main.readData("data.csv");
        main.printList();
    }

    // -----------------READ DATA----------------------------------//
    public void readData(String filename) {
        int salesNumber = 1;
        try {
            Scanner data = new Scanner(new File(filename));
            while (data.hasNext()) {
                String[] line = data.nextLine().split(",");
                if (line[0].equals("R")) {
                    Widget w = new Widget(Integer.parseInt(line[1]), Double.parseDouble(line[2]));
                    widgets.add(w);
                    System.out.println();
                    w.display();
                } else if (line[0].equals("P")) {
                    activateDiscount(Double.parseDouble(line[1]));
                } else if (line[0].equals("S")) {
                    System.out.print("\nSales No " + salesNumber++);
                    System.out.println("| Order for " + Integer.parseInt(line[1]) + " widgets:");
                    sellWidgets(widgets, Integer.parseInt(line[1]), promotion);
                } else
                    System.out.println("Invalid card number");
            }
        } catch (FileNotFoundException e) {
            System.out.println("Missing file " + e.getMessage());
        } catch (NumberFormatException e) {
```

```
            System.out.println("invalid number");
        } catch (Exception e) {
            System.out.println("An error has occured " + e.getMessage());
        }
    }

    // -----------------SELL WIDGETS---------------------------------//
    public void sellWidgets(LinkedList<Widget> widgets, int amount, boolean cardPromotion) {
        NumberFormat nf = NumberFormat.getCurrencyInstance();
        NumberFormat pf = NumberFormat.getPercentInstance();
        int originalAmount = amount; // if remains unchanged, order was completely unfulfilled
        double customerPrice = 0;
        int qty = 0; // to represent quantity in particular widget shipment
        double markup = .3;
        double cost = 0; // per widget shipment
        double total = 0; // represents total order

        Iterator<Widget> itr = widgets.listIterator();
        while (itr.hasNext() && amount > 0) {
            Widget w = itr.next();
            qty = w.getQuantity();
            customerPrice = (markup * w.getPrice()) + w.getPrice();
            if (qty >= amount) { // if shipment contains enough to fill order
                w.updateQuantity(qty - amount); // reduce shipment's inventory by this amount
                cost = amount * customerPrice;
                total = total + cost;
                System.out.println( amount + " widgets at " + nf.format(customerPrice) + " each
Sales: " + nf.format(cost));
                amount = amount - qty;
                if (amount == 0) {
                    itr.remove();
                }
                break; // if sold all items, break out of iteration

            } else { // if there isn't enough, sell whatever is currently in stock, then adjust
the amount accordingly. Still iterating the list, but now with a new amount
                amount = amount - qty;
                cost = qty * customerPrice;
                total = total + cost;
                System.out.println(qty + " widgets at " + nf.format(customerPrice) + " each
Sales: " + nf.format(cost));
                itr.remove();
            }
```

```
        }
        if (amount == originalAmount) {
            System.out.println("Unfortunately, we are completely sold out at this time. ");
        } else {
            if (cardPromotion) {
                System.out.println("Promotional discount of " + pf.format(discount) + " has been
applied.");
                count++;
                total = total - (discount * total);
                if (count > 1)
                    deactivateDiscount();
            }
            if (amount > 0) {
                System.out.println("Unfortunately, " + amount + " widgets are unavailable.");
            }
        }
        System.out.println("TOTAL SALES: " + nf.format(total));
    }

    // -----------------ACTIVATE DISCOUNT---------------------------------//
    public static void activateDiscount(double value) {
        promotion = true;
        discount = value;
        count = 0;
        System.out.println("\n**Promotion has been activated for next two orders.**");
    }
    // -----------------DEACTIVATE DISCOUNT---------------------------------//
    public static void deactivateDiscount() {
        promotion = false;
        discount = 0;
        count = 0;
    }
    // -----------------PRINT LIST---------------------------------//
    public void printList() {
        Iterator<Widget> itr = widgets.listIterator();
        System.out.println("\n-----------------------------------");
        System.out.println("Widgets still in stock:");
        while (itr.hasNext()) {
            Widget w = itr.next();
            w.display();
        }
    }
}// Main
```