

CMPT220 - Program 9

Program Due: Monday, April 20th, before noon. (saved, submitted and printed)

Name the project **Prog9YourLastName**

Name the main class **ShoppingDemoYourLastName.java**

Name the Item class **ItemYourLastName.java**

Name the Node class **NodeYourLastName**

Name the LinkedList class **KeyedListYourLastName.java**

It's time to go shopping again, but with a twist! Each **Item** in the list has a name, a quantity and a price. Luckily, you have most of the code available to you from your wonderful CMPT220 class. (Of course, you don't mention this to your new boss – let her be impressed by how quickly you complete this project.)

This program lets us again exercise the Keyed List Class. You are to complete the **LinkedList implementation** in which the **add** method inserts an item into its proper sorted location in the list.

Items on our list will again consist of the name of the item (as the key), the quantity of that item and the item's unit price. The key field is implemented using the Java String class. All key comparisons should be done in a case insensitive manner using the **String** class comparison methods **equalsIgnoreCase** and **compareToIgnoreCase**.

Your program will again be menu-driven and will allow the shopper to perform the following functions, each corresponding to a method of the Keyed List Class.

The **KeyedListYourLastName** class will implement these public methods of the linkedList version of the Keyed List Class

public KeyedListYourLastName(): Default constructor creates an empty list. Initialize the private attribute myHead to null.

public void clear(): Resets the list to the empty state.

public boolean add(Item product): Inserts the product into the list if its key is unique (case insensitive). Returns true if successful; otherwise, false. Your add method must place the new item where it belongs in the list.

public boolean remove(String keyValue): Deletes the Item with the given key value (the name attribute). Returns true if an Item was found and deleted; else, false.

public ItemYourLastName retrieve(String keyValue): Returns the **ItemYourLastName** having the specified key value; otherwise, return null if not found.

public boolean isEmpty(): Returns true if and only if the list is empty.

public boolean isFull(): Returns false (since the list can never be full).

public void print(): Prints all of the ItemYourLastNames in the list in ascending order by the key field.

Include the following methods in your **KeyedListYourLastName** class:

public int getCount(): Returns the total number of ItemYourLastNames to be purchased by accumulating the sum of the quantity fields of all ItemYourLastNames in the list.

public double calcTotal(): Returns the total cost of all ItemYourLastNames in the list. For each ItemYourLastName multiply the unit price by the quantity.

(Notice that these method headers are exactly the same as they were in Prog8!)

You'll also need to implement a **NodeYourLastName** class that contains an **ItemYourLastName** (called **myData**) and a reference (called **myNext**) to connect to the next **NodeYourLastName** in the linked list.

The program's menu should appear as shown below. Please **order and number** the menu choices as shown below.

1. Add an item to the list
2. Delete an item from the list
3. Print each item in the list
4. Search for a user-specified item in the list
5. Count the total number of items in the list
6. Total the cost of the items in the list
7. Determine whether the list is empty
8. Determine whether the list is full
9. Clear the list
0. Quit

Each menu command should display informative prompt messages as well as print a message indicating the result of processing that command. See the sample program run for example prompt and output messages.

We will again use an input file to get the initial data for this program. The input file will contain a line that specifies the number of ItemYourLastNames in the file, followed by the Name, Quantity and Price of each Item, each input value on a line by itself. Here is an example of what the input file will look like:

```
2
book
5
15.00
dvd
8
20.00
```

Your program should start by asking for the name of the initial input file, then reading in and adding those ItemYourLastNames to the KeyedListYourLastName. After that, present the menu -- all of the rest of the data will be entered via the keyboard.

Be sure to submit your input file(s) – the easiest way to do this is to copy your input file(s) into the project directory before you .zip it.

NOTE: This program reuses LOTS of your prog8 code! You might want to start by making a copy of your entire prog8 project, then replacing code as needed. The easiest way to do this is to follow these steps.

In Eclipse,
find your project in the Package Explorer,
right-click it and choose Copy.

Then right-click again and choose Paste.

When it asks for a name, type in **Prog9YourLastName**. Don't forget to update documentation, etc.!

That's it – you now have a copy of your code. You'll need to completely change the code in the class **KeyedListYourLastName**, of course, and add the class **NodeYourLastName**, but the rest of the code should still work, if it was well-designed for prog8.

Sample Program Run (assume the input file from above)

Please enter the path and name for the data file: **C:\\Users\\Public\\prog9input.txt**
Thank you.

<prints menu>

Please enter your choice: **1**

Please provide information about the item to add.

Please enter the name: **Cereal**

Please enter the quantity: **4**

Please enter the unit price: **4.00**

Cereal has been added to the cart.

<prints menu>

Please enter your choice: **1**

Please provide information about the item to add.

Please enter the name: **Bananas**

Please enter the quantity: **3**

Please enter the unit price: **0.80**

Bananas has been added to the cart.

<prints menu>

Please enter your choice: **5**

Your cart contains a total of 20 items.

<prints menu>

Please enter your choice: **6**

The total price of all of your items is \$253.40.

<prints menu>

Please enter your choice: **4**

Please enter the name of the item to find in your cart: **cereal**

Yes, you have 4 Cereal at \$4.00 each.

<prints menu>

Please enter your choice: **2**

Please enter the name of the item to delete from your cart: **pizza**

Sorry, but your cart does not contain the item pizza.