

<p>Here is where we left off in Exercise 6</p>	<p>An Item class An ItemList class A GoShopping class</p> <p>GoShopping has a main method. When executed it prompts for a number of items, saves the item name, cost, and quantity in an array of Items and prints the names of each Item in the array.</p>
<p>Item</p>	<pre>package com.cosc210; public class Item { private String name = "default"; private double cost = 0.0; private int quantity = 0; public Item(){ } public String getName() { return name; } public void setName(String name) { this.name = name; } public double getCost() { return cost; } public void setCost(double cost) { this.cost = cost; } public int getQuantity() { return quantity; } public void setQuantity(int quantity) { this.quantity = quantity; } }</pre>
<p>ItemList</p>	<pre>package com.cosc210; import java.util.Scanner; public class ItemList { Item items[] = null; public ItemList(){ } public ItemList(int numberOfItems){ items = this.createList(numberOfItems); this.displayItems(items); } public void addItem(Item itemIn, int index){ if (items[index].equals(null)){ System.out.println("adding Item to an empty slot in the array"); items[index] = itemIn; } else { System.out.println("replacing item " + items[index].getName()); System.out.println("with " + itemIn.getName()); items[index] = itemIn; } } }</pre>

```
    }  
}  
  
public void deleteItem(Item itemIn, int index){  
    if (items[index].getName().equals(itemIn.getName())){  
        System.out.println(items[index].getName() + " is  
            deleted from the list");  
        items[index] = null;  
    } else {  
        System.out.println(items[index].getName() + " is in  
            the list at the supplied index");  
        System.out.println(itemIn.getName() + " does not  
            match what is in the list");  
        System.out.println("nothing deleted");  
    }  
}  
  
public void displayItems(Item list[]){  
    int sizeOfList = list.length;  
    for (int n=0;n<sizeOfList;n++){  
        System.out.println("Item is " + list[n].getName());  
    }  
}  
  
public Item[] createList(int listSize) {  
    Item currentItems[] = new Item[listSize];  
  
    Scanner inputName = new Scanner(System.in);  
    Scanner inputCost = new Scanner(System.in);  
    Scanner inputQuantity = new Scanner(System.in);  
  
    for (int x=0;x<listSize;x++){  
        currentItems[x] = new Item();  
        System.out.println("Enter an item name: ");  
  
        currentItems[x].setName(inputName.next());  
        System.out.println("Enter item cost: ");  
  
        currentItems[x].setCost(inputCost.nextDouble());  
        System.out.println("Enter quantity of this item: ");  
  
        currentItems[x].setQuantity(inputQuantity.nextInt());  
    }  
    return currentItems;  
}  
}
```

GoShopping	<pre>package com.cosc210; import java.util.Scanner; public class GoShopping { public static void main(String[] args) { Scanner input = new Scanner(System.in); System.out.println("Enter how many items you will be buying today: "); int numberOfItems = input.nextInt(); ItemList myList = new ItemList(numberOfItems); } }</pre>
Output from the execution of GoShopping	<pre>Enter how many items you will be buying today: 4 Enter an item name: milk Enter item cost: 2.95 Enter quantity of this item: 2 Enter an item name: bread Enter item cost: 2.35 Enter quantity of this item: 2 Enter an item name: eggs Enter item cost: 1.38 Enter quantity of this item: 3 Enter an item name: sugar Enter item cost: 3.78 Enter quantity of this item: 1 Item is milk Item is bread Item is eggs Item is sugar</pre>

<p>Use the addItem method in the GoShopping client</p> <p>The addItem method takes two arguments: an Item and an index into the Items array</p> <p>For this example we will use a random number to choose the index into the array of items to pick the item to replace when we manually add a new item to the list</p> <p>GoShopping still creates the list of Items before trying to use the addItem method</p>	<pre>package com.cosc210; import java.util.Scanner; import java.util.Random; public class GoShopping { public static void main(String[] args) { Scanner input = new Scanner(System.in); System.out.println("Enter how many items you will be buying today: "); int numberOfItems = input.nextInt(); ItemList myList = new ItemList(numberOfItems); Item newItem = new Item(); newItem.populateItem(); Random myRandom = new Random(); myList.addItem(newItem, myRandom.nextInt(numberOfItems)); } }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Create the populateItem() method we used in the previous update of GoShopping

```
package com.cosc210;
import java.util.Scanner;
public class Item {
    private String name = "default";
    private double cost = 0.0;
    private int quantity = 0;

    public Item(){
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getCost() {
        return cost;
    }
    public void setCost(double cost) {
        this.cost = cost;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    public void populateItem(){
        Scanner inputName = new Scanner(System.in);
        Scanner inputCost = new Scanner(System.in);
        Scanner inputQuantity = new Scanner(System.in);

        System.out.println("Enter an item name: ");
        this.setName(inputName.next());
        System.out.println("Enter item cost: ");

        this.setCost(inputCost.nextDouble());
        System.out.println("Enter quantity of this item: ");

        this.setQuantity(inputQuantity.nextInt());
    }
}
```

<p>Output from using the populateItem() method in our client called GoShopping.</p>	<pre>Enter how many items you will be buying today: 4 Enter an item name: bread Enter item cost: 2.38 Enter quantity of this item: 3 Enter an item name: milk Enter item cost: 3.45 Enter quantity of this item: 2 Enter an item name: ice Enter item cost: 1.19 Enter quantity of this item: 4 Enter an item name: oj Enter item cost: 3.78 Enter quantity of this item: 2 Item is bread Item is milk Item is ice Item is oj Enter an item name: ← this is beginning of the cake frosting populateItem call Enter item cost: 3.12 Enter quantity of this item: 712 replacing item milk ← this is the addItem call with cake</pre>
-------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

How else could we create and populate an Item to use in the addItem method?

Instead of using a method to prompt for and create an item and its state information, we could use an Item constructor to take the state information as parameters. Use these parameters and the setters within an item to fill in its state information.

```
package com.cosc210;
import java.util.Scanner;
public class Item {
    private String name = "default";
    private double cost = 0.0;
    private int quantity = 0;

    public Item(){
    }

    public Item(String name, double cost, int quantity){
        this.setName(name);
        this.setCost(cost);
        this.setQuantity(quantity);
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getCost() {
        return cost;
    }
    public void setCost(double cost) {
        this.cost = cost;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    public void populateItem(){
        Scanner inputName = new Scanner(System.in);
        Scanner inputCost = new Scanner(System.in);
        Scanner inputQuantity = new Scanner(System.in);

        System.out.println("Enter an item name: ");
        this.setName(inputName.next());
        System.out.println("Enter item cost: ");

        this.setCost(inputCost.nextDouble());
        System.out.println("Enter quantity of this item: ");

        this.setQuantity(inputQuantity.nextInt());
    }
}
```

<p>Let us alter GoShopping to use the new constructor we created.</p>	<pre>package com.cosc210; import java.util.Scanner; import java.util.Random; public class GoShopping { public static void main(String[] args) { Scanner input = new Scanner(System.in); System.out.println("Enter how many items you will be buying today: "); int numberOfItems = input.nextInt(); ItemList myList = new ItemList(numberOfItems); Item newItem = new Item(); newItem.populateItem(); Random myRandom = new Random(); myList.addItem(newItem, myRandom.nextInt(numberOfItems)); Scanner inputName = new Scanner(System.in); Scanner inputCost = new Scanner(System.in); Scanner inputQuantity = new Scanner(System.in); String name = null; double cost = 0.0; int quantity = 0; System.out.println("Enter an item name: "); name = inputName.next(); System.out.println("Enter item cost: "); cost = inputCost.nextDouble(); System.out.println("Enter quantity of this item: "); quantity = inputQuantity.nextInt(); Item anotherItem = new Item(name, cost, quantity); myList.addItem(anotherItem, myRandom.nextInt(numberOfItems)); } }</pre>
<p>Output from the changes we just made</p>	<pre>Enter how many items you will be buying today: 4 Enter an item name: bread Enter item cost: 2.38 Enter quantity of this item: 3 Enter an item name: milk Enter item cost: 3.45 Enter quantity of this item: 2</pre>

	<pre> Enter an item name: ice Enter item cost: 1.19 Enter quantity of this item: 4 Enter an item name: oj Enter item cost: 3.78 Enter quantity of this item: 2 Item is bread Item is milk Item is ice Item is oj Enter an item name: ← populateItem method call cake frosting Enter item cost: 3.12 Enter quantity of this item: 712 replacing item milk ← addItem call with cake Enter an item name: ← prep so we can use the Bread new constructor Enter item cost: 2.35 Enter quantity of this item: 4 replacing item bread ← addItem call after the with bread constructor has been run </pre>
<p>What if we want to delete an item from the list, we do have a deleteItem method.</p> <p>What will it take to use it?</p> <p>deleteItem takes an Item and an index as parameters, checks to see that the item we pass in is the same as the item we want to delete. If it is the same deletes the item.</p>	<pre> package com.cosc210; import java.util.Scanner; import java.util.Random; public class GoShopping { public static void main(String[] args) { Scanner input = new Scanner(System.in); System.out.println("Enter how many items you will be buying today: "); int numberOfItems = input.nextInt(); ItemList myList = new ItemList(numberOfItems); Item newItem = new Item(); newItem.populateItem(); Random myRandom = new Random(); myList.addItem(newItem, myRandom.nextInt(numberOfItems)); </pre>

	<pre>Scanner inputName = new Scanner(System.in); Scanner inputCost = new Scanner(System.in); Scanner inputQuantity = new Scanner(System.in); String name = null; double cost = 0.0; int quantity = 0; System.out.println("Enter an item name: "); name = inputName.next(); System.out.println("Enter item cost: "); cost = inputCost.nextDouble(); System.out.println("Enter quantity of this item: "); quantity = inputQuantity.nextInt(); Item anotherItem = new Item(name, cost, quantity); myList.addItem(anotherItem, myRandom.nextInt(numberOfItems)); int z=0; while (z<numberOfItems){ myList.deleteItem(anotherItem, z); z++; } }</pre>
<p>Output from looping through the list of Items searching for the item to delete</p>	<pre>Enter how many items you will be buying today: 4 Enter an item name: bread Enter item cost: 2.38 Enter quantity of this item: 3 Enter an item name: milk Enter item cost: 3.45 Enter quantity of this item: 2 Enter an item name: ice Enter item cost: 1.19 Enter quantity of this item: 4 Enter an item name: oj Enter item cost: 3.78 Enter quantity of this item: 2 Item is bread Item is milk Item is ice Item is oj</pre>

	<pre> Enter an item name: cake frosting Enter item cost: 3.12 Enter quantity of this item: 712 replacing item milk with cake Enter an item name: bread Enter item cost: 2.35 Enter quantity of this item: 4 replacing item bread with bread bread is deleted from the list cake is in the list at the supplied index bread does not match what is in the list nothing deleted ice is in the list at the supplied index bread does not match what is in the list nothing deleted oj is in the list at the supplied index bread does not match what is in the list nothing deleted </pre>
<p>How could we use this program we have created to either write to a file or read from a file?</p> <p>Ideas for behaviors: printListToFile() updateListInFile() readExistingList()</p> <p>This step will implement a version of printListToFile()</p>	<pre> package com.cosc210; import java.io.FileOutputStream; import java.io.PrintStream; import java.util.Scanner; public class ItemList { Item items[] = null; public ItemList(){ } public ItemList(int numberOfItems){ items = this.createList(numberOfItems); this.displayItems(items); } public void addItem(Item itemIn, int index){ if (items[index].equals(null)){ System.out.println("adding Item to an empty slot in the array"); items[index] = itemIn; } else { System.out.println("replacing item " + items[index].getName()); System.out.println("with " + itemIn.getName()); items[index] = itemIn; } } } </pre>

```
}  
}  
  
public void deleteItem(Item itemIn, int index){  
    if (items[index].getName().equals(itemIn.getName())){  
        System.out.println(items[index].getName() + " is  
            deleted from the list");  
        items[index] = null;  
    } else {  
        System.out.println(items[index].getName() + " is in  
            the list at the supplied index");  
        System.out.println(itemIn.getName() + " does not  
            match what is in the list");  
        System.out.println("nothing deleted");  
    }  
}  
  
public void displayItems(Item list[]){  
    int sizeOfList = list.length;  
    for (int n=0;n<sizeOfList;n++){  
        System.out.println("Item is " + list[n].getName());  
    }  
}  
  
public Item[] createList(int listSize) {  
    Item currentItems[] = new Item[listSize];  
  
    Scanner inputName = new Scanner(System.in);  
    Scanner inputCost = new Scanner(System.in);  
    Scanner inputQuantity = new Scanner(System.in);  
  
    for (int x=0;x<listSize;x++){  
        currentItems[x] = new Item();  
        System.out.println("Enter an item name: ");  
  
        currentItems[x].setName(inputName.next());  
        System.out.println("Enter item cost: ");  
  
        currentItems[x].setCost(inputCost.nextDouble());  
        System.out.println("Enter quantity of this item: ");  
  
        currentItems[x].setQuantity(inputQuantity.nextInt());  
    }  
    return currentItems;  
}  
  
public void printListToFile(){  
    FileOutputStream out; // declare a file output object  
    PrintStream p; // declare a print stream object  
    try{  
        out = new FileOutputStream("C:\\\\theList.txt");  
        p = new PrintStream( out );  
  
        int listSize = items.length;  
        for (int f=0;f<listSize;f++){  
            if (items[f] != null) {  
                System.out.println(f + "about to write to file");  
            }  
        }  
    }  
}
```

	<pre>p.println(items[f].getName()); p.println(items[f].getCost()); p.println(items[f].getQuantity()); System.out.println("done with element " + f); } } p.close(); } catch (Exception e){ System.err.println ("Error writing to file"); e.printStackTrace(); } } }</pre>
<p>Update the GoShopping client to use the printListToFile() method</p>	<pre>package com.cosc210; import java.util.Scanner; import java.util.Random; public class GoShopping { public static void main(String[] args) { Scanner input = new Scanner(System.in); System.out.println("Enter how many items you will be buying today: "); int numberOfItems = input.nextInt(); ItemList myList = new ItemList(numberOfItems); Item newItem = new Item(); newItem.populateItem(); Random myRandom = new Random(); myList.addItem(newItem, myRandom.nextInt(numberOfItems)); Scanner inputName = new Scanner(System.in); Scanner inputCost = new Scanner(System.in); Scanner inputQuantity = new Scanner(System.in); String name = null; double cost = 0.0; int quantity = 0; System.out.println("Enter an item name: "); name = inputName.next(); System.out.println("Enter item cost: "); cost = inputCost.nextDouble(); System.out.println("Enter quantity of this item: "); quantity = inputQuantity.nextInt(); Item anotherItem = new Item(name, cost, quantity); myList.addItem(anotherItem, myRandom.nextInt(numberOfItems)); int z=0; while (z<numberOfItems){</pre>

	<pre> myList.deleteItem(anotherItem, z); z++; } myList.printListToFile(); } }</pre>
<p>Output from running the GoShopping class with our new line of code that invokes the printListToFile() method</p>	<pre>Enter how many items you will be buying today: 3 Enter an item name: item1 Enter item cost: 1 Enter quantity of this item: 1 Enter an item name: item2 Enter item cost: 2 Enter quantity of this item: 2 Enter an item name: item3 Enter item cost: 3 Enter quantity of this item: 3 Item is item1 Item is item2 Item is item3 Enter an item name: item4 Enter item cost: 4 Enter quantity of this item: 4 replacing item item1 with item4 Enter an item name: item5 Enter item cost: 5 Enter quantity of this item: 5 replacing item item4 with item5 item5 is deleted from the list item2 is in the list at the supplied index item5 does not match what is in the list nothing deleted item3 is in the list at the supplied index item5 does not match what is in the list</pre>

	<pre>nothing deleted labout to write to file done with element 1 2about to write to file done with element 2</pre>
Contents of C:\theList.txt	<pre>item2 2.0 2 item3 3.0 3</pre>
BONUS POINTS	<ol style="list-style-type: none">1) Run this program in many ways and record what errors you find.2) Suggest possible solutions to the errors you find.3) There are issues with writing to the file, describe them.4) If we wanted to delete from the file when we deleted from the list how might we do that?