

<p>Here is where we left off in Exercise 7</p>	<p><b>An Item class</b> <b>An ItemList class</b> <b>A GoShopping class</b></p> <p>GoShopping has a main method. When executed:</p> <ol style="list-style-type: none"><li>1) it prompts for a number of items,</li><li>2) saves the item name, cost, and quantity in an array of Items and</li><li>3) prints the names of each Item in the array.</li><li>4) Next it creates a new Item object and</li><li>5) invokes the populateItem() method to give it state.</li><li>6) We put the item in a random spot in the array of Items.</li><li>7) We input the name, cost, and quantity for a new Item</li><li>8) and create the item object passing the inputted values into the constructor that accepts a String, a double, and an int.</li><li>9) add the new item to a random spot in the array of Items</li><li>10) loop through the items of the array and delete the new item</li><li>11) Print the contents of the array to a file</li></ol>
<p>Item</p>	<pre>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);</pre>

	<pre>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()); } }</pre>
ItemList	<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;         }     }      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&lt;sizeOfList;n++){             System.out.println("Item is " + list[n].getName());         }     } }</pre>

	<pre>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&lt;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 {         // Create a new file output stream         // connected to "myfile.txt"         out = new FileOutputStream("C:\\\\theList.txt");          // Connect print stream to the output stream         p = new PrintStream( out );          int listSize = items.length;         for (int f=0;f&lt;listSize;f++){             if (items[f] != null) {                 System.out.println(f + "about to write to file");                 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>
GoShopping	<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);</pre>

	<pre>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&lt;numberOfItems){   myList.deleteItem(anotherItem, z);   z++; } myList.printListToFile(); }</pre>
<p>Output from the execution of 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: 2 Enter an item name: milk Enter item cost: 3.45 Enter quantity of this item: 1 Enter an item name:</pre>

	<pre> eggs Enter item cost: 2.18 Enter quantity of this item: 2 Enter an item name: cheese Enter item cost: 3.78 Enter quantity of this item: 1 Item is bread Item is milk Item is eggs Item is cheese Enter an item name: bacon Enter item cost: 4.18 Enter quantity of this item: 4 replacing item milk with bacon Enter an item name: syrup Enter item cost: 3.10 Enter quantity of this item: 1 replacing item cheese with syrup bread is in the list at the supplied index syrup does not match what is in the list nothing deleted bacon is in the list at the supplied index syrup does not match what is in the list nothing deleted eggs is in the list at the supplied index syrup does not match what is in the list nothing deleted syrup is deleted from the list 0about to write to file done with element 0 1about to write to file done with element 1 2about to write to file done with element 2 </pre>
<p>Let us now analyze what we have written and fix whatever errors we find</p>	
<p>What happens if we put a character or word in when it is expecting a number</p>	<pre> Enter how many items you will be buying today: none Exception in thread "main" java.util.InputMismatchException at java.util.Scanner.throwFor(Unknown Source) at java.util.Scanner.next(Unknown Source) at java.util.Scanner.nextInt(Unknown Source) </pre>

	<pre>at java.util.Scanner.nextInt(Unknown Source) at com.cosc210.GoShopping.main(GoShopping.java:10)</pre>
<p>There is a bunch of information in the exception that occurs</p>	<p>1) <a href="#">java.util.InputMismatchException</a> The exception that was thrown is given, we can use this to create a try catch set of statements or we can use a generic Exception instead of the InputMismatchException or have more than one catch statement.</p> <p>2) <a href="#">at com.cosc210.GoShopping.main(GoShopping.java:10)</a> The source file and line of code in the file where the source of the exception occurred. It may be around this line of code that the error occurred.</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(); // ← line 10</pre>
<p>Let us put a try and catch around line 10</p> <p>You have to declare the int numberOfItems outside the try block</p> <p>Add a try and catch using the exception we witnessed in the previous section of this document</p>	<pre>package com.cosc210; import java.util.Scanner; import java.util.Random; import java.util.InputMismatchException;  public class GoShopping {     public static void main(String[] args) {         Scanner input = new Scanner(System.in);         int numberOfItems = 0;         System.out.println("Enter how many items you will be         buying today: ");         try {             numberOfItems = input.nextInt();         } catch (InputMismatchException ime){             System.out.println("You did not enter an appropriate             value:");         }          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);</pre>

	<pre>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&lt;numberOfItems){     myList.deleteItem(anotherItem, z);     z++; } myList.printListToFile(); }</pre>
<p>Output from executing Go shopping with the try and catch inside</p> <p>The error says ‘n must be positive’</p> <p>We initialized</p>	<pre>Enter how many items you will be buying today: n You did not enter an appropriate value: Enter an item name: a Enter item cost: 1 Enter quantity of this item: 1 Exception in thread "main" java.lang.IllegalArgumentException: n must be positive     at java.util.Random.nextInt(Unknown Source)     at com.cosc210.GoShopping.main(GoShopping.java:40)</pre>
<p>Let us fix the ‘n must be positive’ error</p>	<pre>package com.cosc210; import java.util.Scanner; import java.util.Random; import java.util.InputMismatchException; public class GoShopping {     public static void main(String[] args) {         Scanner input = new Scanner(System.in);         int numberOfItems = 0;         System.out.println("Enter how many items you will be             buying today: ");         try {             numberOfItems = input.nextInt();         } catch (InputMismatchException ime){             System.out.println("You did not enter an appropriate                 value:");             System.out.println("Number of items defaulted to 2");             numberOfItems = 2;         }         ItemList myList = new ItemList(numberOfItems);  #### the rest is the same - no changes ####</pre>

<p>We print the message and set the value to an acceptable amount in the code</p> <p>Then move on</p>	<pre> Enter how many items you will be buying today: n You did not enter an appropriate value: Number of items defaulted to 2 Enter an item name: a Enter item cost: 1 Enter quantity of this item: 1 Enter an item name: b Enter item cost: 3 Enter quantity of this item: 3 Item is a Item is b Enter an item name: f Enter item cost: 3 Enter quantity of this item: 3 replacing item a with f Enter an item name: t Enter item cost: 5 Enter quantity of this item: 5 replacing item f with t t is deleted from the list b is in the list at the supplied index t does not match what is in the list nothing deleted labout to write to file done with element 1 </pre>
<p>What else might we do that is a bit more robust?</p> <p>Let's fix it so this code asks again if an exception occurs.</p> <ol style="list-style-type: none"> <li>1) we take in the input as a String so anything the user enters is accepted</li> <li>2) put the try catch inside a while</li> </ol>	<pre> package com.cosc210; import java.util.Scanner; import java.util.Random; import java.util.InputMismatchException; public class GoShopping {     public static void main(String[] args) {         Scanner input = new Scanner(System.in);         int numberOfItems = 0;         String inValue = "0";         Integer wrapInt = new Integer(inValue);         boolean valid = false;          System.out.println("Enter how many items you will be         buying today: ");         while ( valid == false ){             try {                 inValue = input.next(); </pre>

<p>loop</p> <p>3) convert the in value String to an Integer</p> <p>4) if the number format exception does not occur we drop out of the while loop, if it does occur we process the catch</p> <p>5) no exception save the Integer as an int and move on</p>	<pre>// input all values as a String wrapInt = new Integer(inValue); // this statement only executes if the string // entered does not cause a NumberFormatException valid = true; } catch (NumberFormatException nfe){     System.out.println( nfe.getMessage()         + " is not a valid format for an integer." );     System.out.println("You did not enter an         appropriate value, try again:"); } numberOfItems = wrapInt; }</pre> <p>#### no more changes ####</p>
<p>Output from the newly created code</p>	<pre>Enter how many items you will be buying today: N For input string: "N" is not a valid format for an integer. You did not enter an appropriate value, try again: a For input string: "a" is not a valid format for an integer. You did not enter an appropriate value, try again: 2 Enter an item name: bread Enter item cost: 2.12 Enter quantity of this item: 3 Enter an item name: eggs Enter item cost: 2.12 Enter quantity of this item: 5 Item is bread Item is eggs Enter an item name: sugar Enter item cost: 3.12 Enter quantity of this item: 4 replacing item bread with sugar Enter an item name: Jello Enter item cost: 0.34 Enter quantity of this item: 45 replacing item sugar with Jello Jello is deleted from the list eggs is in the list at the supplied index</pre>

	<pre>Jello does not match what is in the list nothing deleted labout to write to file done with element 1</pre>
<p>Where else might we break this code? Here is an example.</p> <p>Where does this need fixing?</p> <p>Fix this error either by printing a message and setting a default or asking for a new value.</p>	<pre>Enter how many items you will be buying today: 1 Enter an item name: bread 234 alpha george Enter item cost: twelve Exception in thread "main" java.util.InputMismatchException     at java.util.Scanner.throwFor(Unknown Source)     at java.util.Scanner.next(Unknown Source)     at java.util.Scanner.nextDouble(Unknown Source)     at com.cosc210.ItemList.createList(ItemList.java:57)     at com.cosc210.ItemList.&lt;init&gt;(ItemList.java:12)     at com.cosc210.GoShopping.main(GoShopping.java:44)</pre>