

```

1 // foj/4/estore/ShoppingCart.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 /**
7  * A ShoppingCart keeps track of a customer's purchases.
8  *
9  * @see EStore
10 * @version 4
11 */
12
13 public class ShoppingCart
14 {
15     /** replace these two fields by a single ArrayList
16     private int count; // number of Items in this ShoppingCart
17     private int cost; // total cost of Items in this ShoppingCart
18
19     /**
20     * Construct a new empty ShoppingCart.
21     */
22
23     public ShoppingCart()
24     {
25         count = 0;
26         cost = 0;
27     }
28
29     /**
30     * Add an Item to this ShoppingCart.
31     *
32     * @param item the Item to add.
33     */
34
35     public void addItem( Item item )
36     {
37         /** this code just keeps track of the totals
38         /** replace it with code that adds the item to the list
39         count++;
40         this.cost += item.getCost(); // Java idiom: a += b means a = a +
41         }
42
43     /**
44     * Return an Item from this ShoppingCart.
45     *
46     * @param item the Item to return.
47     */
48
49     public void returnItem( Item item )
50     {
51         /** look through the list looking for Item
52         /** remove it if it's there
53         }
54
55     /**
56     * What happens when this ShoppingCart is asked how many

```

```

57
58     * Items it contains.
59     * @return the number of items in this ShoppingCart.
60     */
61
62     public int getCount()
63     {
64         /** get this information from the list,
65         /** since the count field no longer exists
66         return count;
67     }
68
69     /**
70     * What happens when this ShoppingCart is asked the total
71     * cost of the Items it contains.
72     *
73     * @return the total cost of the items in this ShoppingCart.
74     */
75     public int getCost()
76     {
77         /** get this information from the list,
78         /** since the cost field no longer exists
79         return cost;
80     }
81
82     /**
83     * Write the contents of this ShoppingCart to a Terminal.
84     *
85     * @param t the Terminal to use for output.
86     */
87
88     public void showContents( Terminal t )
89     {
90         /** work to do here ...
91         t.println(" [sorry, can't yet print ShoppingCart contents]");
92     }
93 }

```