

```

1 // foj/1/estore/ESTore.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 /**
7  * An EStore object simulates the behavior of a simple on line
8  * shopping web site.
9
10 * It contains a Terminal object to model the customer's browser
11 * and several Item objects a customer can add to her ShoppingCart.
12
13 * @version 1
14 */
15
16 public class EStore
17 {
18     private String storeName = "Virtual Minimal Minimal";
19
20     // Use a Terminal object to communicate with customers.
21     private Terminal browser = new Terminal();
22
23     // The store stocks two kinds of Items.
24     private Item widget = new Item(10); // widgets cost $10
25     private Item gadget = new Item(13); // gadgets cost $13
26
27     private String selectionList = "(gadget, widget, checkout)";
28
29     /**
30      * Visit this EStore.
31
32      * Loop allowing visitor to select items to add to her
33      * ShoppingCart.
34      */
35
36     public void visit()
37     {
38         // Create a new, empty ShoppingCart.
39         ShoppingCart basket = new ShoppingCart();
40
41         // Print a friendly welcome message.
42         browser.println("Welcome to " + storeName );
43
44         // Change to false when customer is ready to leave:
45         boolean stillShopping = true;
46
47         while ( stillShopping ) {
48             Item nextPurchase = selectItem();
49             if ( nextPurchase == null ) {
50                 stillShopping = false;
51             }
52             else {
53                 basket.add( nextPurchase );
54             }
55         }
56         int numberPurchased = basket.getCount();

```

```

57     int totalCost      = basket.getCost();
58     browser.println("We are shipping " + numberPurchased + " Items");
59     browser.println("and charging your account $" + totalCost);
60     browser.println("Thank you for shopping at " + storeName);
61 }
62
63 // Discover what the customer wants to do next:
64 // send browser a message to get customer input
65 // examine response to make a choice
66 // If response makes no sense give customer another chance
67
68     private Item selectItem()
69     {
70         String itemName =
71             browser.readWord("Item " + selectionList + " :");
72
73         if ( itemName.equals("widget") ) {
74             return widget;
75         }
76         else if ( itemName.equals("gadget") ) {
77             return gadget;
78         }
79         else if ( itemName.equals("checkout") ) {
80             return null;
81         }
82         else {
83             browser.println("No item named " +
84                 itemName + "; try again" );
85             return selectItem(); // try again
86         }
87     }
88
89     /**
90      * The EStore simulation program begins here when the user
91      * issues the command <code>java EStore</code>.
92      */
93
94     public static void main( String[] args )
95     {
96         // Print this to simulate delay while browser finds store
97         System.out.println("connecting ...");
98
99         // Create the EStore object.
100        EStore website = new EStore();
101
102        // Visit it.
103        website.visit();
104    } // end of class EStore
105

```