

```

1 // Example 4.3 ArrayListDemo.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 // Tell the java compiler that the ArrayList class is in
7 // the java.util part of the library.
8
9 import java.util.ArrayList;
10
11 // Exercise the most important parts of the ArrayList API.
12 //
13 // %> java ArrayListDemo
14 // Create a list containing three SimpleObjects.
15 // 0 zero
16 // 1 one
17 // 2 two
18 // Replace the object at position 0.
19 // Put a new object at 2 and push the rest along.
20 // Print out the list again.
21 // 0 new zero
22 // 1 one
23 // 2 one point five
24 // 3 two
25
26 public class ArrayListDemo
27 {
28     public static void main( String[] args )
29     {
30         System.out.println("Create a list containing three SimpleObjects.
31
32         // Create a new, empty ArrayList
33         // with the ArrayList constructor.
34         ArrayList myList = new ArrayList();
35
36         // Put three things on it with the add()
37         // method - each add appends to the list.
38         myList.add(new SimpleObject("zero"));
39         myList.add(new SimpleObject("one"));
40         myList.add(new SimpleObject("two"));
41
42         // Print the list with a for loop.
43         // size() method tells how long the list is.
44         // get(int index) method retrieves value stored at position index
45         // The (SimpleObject) cast tells Java what type of thing you got
46         for ( int i = 0; i < myList.size(); i++ ) {
47             SimpleObject foo = (SimpleObject)myList.get(i);
48             System.out.println(i + "\t" + foo.name);
49         }
50
51         // set(int index) method changes value stored at position index
52         System.out.println("Replace the object at position 0.");
53         myList.set(0, new SimpleObject("new zero"));
54
55         System.out.println("Put a new object at 2 and push the rest along
56         myList.add(2, new SimpleObject("one point five"));

```

```

57
58     System.out.println("Print out the list again.");
59     for ( int i = 0; i < myList.size(); i++ ) {
60         SimpleObject foo = (SimpleObject)myList.get(i); // note cast!
61         System.out.println(i + "\t" + foo.name);
62     }
63
64
65     // This really simple class exists only to provide
66     // things to put in the ArrayList.
67
68     // It's an inner class, declared inside the ArrayListDemo
69     // class, which is its scope.
70
71     // Since it's visible only here, we are using a public
72     // name field rather than a private field and a public
73     // getName()
74
75     private static class SimpleObject {
76
77         public String name;
78
79         public SimpleObject( String name ) {
80             this.name = name;
81         }
82     } // end of body of inner class SimpleObject
83
84 } // end of body of ArrayList Demo

```