```
 1  // Example 5.3 joi/examples/EqualsDemo.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  // A class illustrating == and equals().
 7  //
 8  // %> java EqualsDemo
 9  // Different objects, same field:
10  // e1 == e1LookAlike ->          false
11  // e1.equals( e1LookAlike ) ->  true
12  // Same object:
13  // e1 == e1Too ->                true
14  //
15  // Different ArrayLists with equal (but not ==) elements:
16  // alist0 == alist1 ->      false
17  // alist0.equals(alist1) -> true
18  //
19  // Different TreeMaps with equal keys
20  // mapping to equal (but !=) values:
21  // tmap0 == tmap1 ->      false
22  // tmap0.equals(tmap1) -> true
23  //
24  // tmap0.toString() -> {sillykey = EqualsDemo value 1}
25  // tmap1.toString() -> {sillykey = EqualsDemo value 1}
26  // Are these Strings == ?    false
27  // Are these Strings equal ? true
28
29  import java.util.ArrayList;
30  import java.util.TreeMap;
31
32  public class EqualsDemo
33  {
34      // Properties of an EqualsDemo object.
35      //
36      // Override equals: two of these objects are equal if
37      // their integer field has the same (i.e. ==) value.
38      //
39      // When you override equals it's customary to override
40      // toString too, so that equal objects return the equal
41      // Strings, so we do.
42
43      private int field;
44
45      public EqualsDemo( int field )
46      {
47          this.field = field;
48      }
49
50      public boolean equals( Object other )
51      {
52          return (other instanceof EqualsDemo)
53              && (this.field == ((EqualsDemo)other).field);
54      }
55
56      public String toString()
```



```
57      {
58          return " EqualsDemo value " + field;
59      }
60
61      public static void main( String[] args )
62      {
63          Terminal t = new Terminal();
64
65          // EqualsDemo object == vs equals()
66          EqualsDemo e1          = new EqualsDemo( 1 );
67          EqualsDemo e1LookAlike = new EqualsDemo( 1 ); // same field.
68          EqualsDemo e1Too       = e1;                  // same object
69
70          t.println("Different objects, same field:");
71          t.println("e1 == e1LookAlike ->         " + (e1 == e1LookAlike));
72          t.println("e1.equals( e1LookAlike ) ->  " + e1.equals( e1LookAlik
73          t.println("Same object:");
74          t.println("e1 == e1Too ->               " + (e1 == e1Too));
75          t.println();
76
77          // Arrays and Maps
78          ArrayList alist0 = new ArrayList();
79          ArrayList alist1 = new ArrayList();
80
81          alist0.add( e1);
82          alist1.add( e1LookAlike );
83
84          t.println( "Different ArrayLists with equal (but not ==) elements
85          t.println( "alist0 == alist1 ->      "  + (alist0 == alist1));
86          t.println( "alist0.equals(alist1) -> "  + alist0.equals(alist1));
87          t.println();
88
89          TreeMap tmap0 = new TreeMap();
90          TreeMap tmap1 = new TreeMap();
91
92          tmap0.put( "sillykey ", e1 );
93          tmap1.put( "sillykey ", e1LookAlike );
94
95          t.println( "Different TreeMaps with equal keys" );
96          t.println( "mapping to equal (but !=) values:" );
97
98          t.println( "tmap0 == tmap1 ->      " + (tmap0 == tmap1));
99          t.println( "tmap0.equals(tmap1) -> " + tmap0.equals(tmap1));
100         t.println();
101
102         // Test Strings for == and equal
103         String s0 = tmap0.toString();
104         String s1 = tmap1.toString();
105         t.println( "tmap0.toString() -> " + s0);
106         t.println( "tmap1.toString() -> " + s1);
107         t.println( "Are these Strings == ?    " + (s0 == s1) );
108         t.println( "Are these Strings equal ? " + (s0.equals(s1)) );
109     }
110 }
```