```
 1  // Example 5.2 joi/examples/OverridingDemo.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  // Small program to illustrate overriding and toString()
 7  //
 8  // Here's what the output looks like:
 9  //
10  // %> java Overriding jessica benjamin
11  // Terminal t = new Terminal();
12  // nobj = new NamedObject( args[0] );
13  // nobj.toString():           jessica
14  // nobj:                      jessica
15  // nobj.toStringfromObject(): NamedObject@206fdf64
16  // nobj = new NamedObject( args[1] );
17  // nobj.toString():           benjamin
18  // nobj:                      benjamin
19  // nobj.toStringfromObject(): NamedObject@2103df64
20  //
21  // toString():                Terminal@201bdf64
22  // t:                         Terminal@201bdf64
23
24  public class OverridingDemo
25  {
26      public static void main( String[] args )
27      {
28          Terminal t = new Terminal();
29          NamedObject nobj;
30
31          t.println("Terminal t = new Terminal();");
32          t.println("nobj = new NamedObject( args[0] );");
33          nobj = new NamedObject( args[0] );
34          t.println( "nobj.toString():           " + nobj.toString() );
35          t.println( "nobj:                      " + nobj );
36          t.println( "nobj.toStringfromObject(): " +
37                     nobj.toStringfromObject());
38
39          t.println("nobj = new NamedObject( args[1] );");
40          nobj = new NamedObject( args[1] );
41          t.println( "nobj.toString():           " + nobj.toString() );
42          t.println( "nobj:                      " + nobj );
43          t.println( "nobj.toStringfromObject(): " +
44                     nobj.toStringfromObject());
45
46          t.println( "\ntoString():               " + t.toString() );
47          t.println( "t:                         " + t );
48      }
49  }
50
51  // A simple class whose instances have one field
52  // and several toString methods. Visible only inside
53  // the OverridingDemo class.
54
55  // you can put two classes in one file as long as only one of them
56  // is public
```



```
57
58  class NamedObject // extends Object, by default
59  {
60      private String name;
61
62      // constructor does the obvious thing
63
64      public NamedObject( String name )
65      {
66          this.name = name;
67      }
68
69      // override toString in class Object
70
71      public String toString()
72      {
73          return name;
74      }
75
76      // access to the overridden method using super
77
78      public String toStringfromObject()
79      {
80          return super.toString();
81      }
82  }
```