Write down what the following code outputs.

```
1 class Utensils                              38     {
2 {                                           39         cost = .99;
3    public double getCost()                  40     }
4    {                                        41     public double getCost()
5        return 0.0;                          42     {
6    }                                        43         return cost;
7    public String latinWord()               44     }
8    {                                        45     public String latinWord()
9        return "sermo";                      46     {
10   }                                        47         return "cochleari";
11   public String use()                      48     }
12   {                                        49     public String use()
13       return "Assist in eating process";  50     {
14   }                                        51         return "Eating Soup Mostly";
15 }                                          52     }
16                                            53 }
17 class Fork extends Utensils                54
18 {                                          55 public class Eating
19    private double cost;                    56 {
20    public Fork()                           57    public static void main(String[] args)
21    {                                       58    {
22        cost = 1.25;                        59      Utensils[] myUtensils = new Utensils[3];
23    }                                       60      myUtensils[0] = new Fork();
24    public double getCost()                 61      myUtensils[1] = new Spoon();
25    {                                       62      System.out.println(
26        return cost;                        63          myUtensils[0].latinWord());
27    }                                       64      myUtensils[0] = new Spoon();
28    public String latinWord()               65      myUtensils[2] = new Fork();
29    {                                       66      for (int x = 0; x < myUtensils.length; x++)
30        return "fuscinulam";                67      {
31    }                                       68        System.out.println(myUtensils[x].use());
32 }                                          69        myUtensils[x] = new Fork();
33                                            70        System.out.println(
34 class Spoon extends Utensils               71            myUtensils[x].getCost());
35 {                                          72      }
36    private double cost;                    73    }
37    public Spoon()                          74 }
```