

```

1 public class animal {
2
3     // Superclass
4     static class Animal {
5
6         // Method Overloading - same name, different
        parameters
7         public void speak() {
8             System.out.println("The animal makes a
            sound.");
9         }
10
11         public void speak(String sound) {
12             System.out.println("The animal says: " +
            sound);
13         }
14
15         public void speak(String sound, int times) {
16             System.out.println("The animal says:");
17             for (int i = 0; i < times; i++) {
18                 System.out.println(sound);
19             }
20         }
21
22         // This method will be overridden
23         public void move() {
24             System.out.println("The animal moves in a
            general way.");
25         }
26     }
27
28     // Subclass
29     static class Dog extends Animal {
30
31         // Method Overriding
32         @Override
33         public void move() {
34             System.out.println("The dog runs on four
            legs.");
35         }
36
37         // Subclass-specific method
38         public void fetch() {
39             System.out.println("The dog fetches the

```

```
39 ball.");
40         }
41     }
42
43     // Main method
44     public static void main(String[] args) {
45         Animal genericAnimal = new Animal();
46         Dog dog = new Dog();
47
48         System.out.println("---- Method Overloading
49         ----");
50         genericAnimal.speak();
51         genericAnimal.speak("Grrr!");
52         genericAnimal.speak("Woof!", 3);
53
54         System.out.println("\n---- Method Overriding
55         ----");
56         genericAnimal.move(); // From Animal
57         dog.move();           // Overridden in Dog
58
59         System.out.println("\n---- Subclass Method
60         ----");
61         dog.fetch();           // Only in Dog
62     }
63 }
```