

## 1 Raining Cats and Dogs

---

Write out the necessary classes for the code below to compile and produce the given output.

OUTPUT:

Dug wolfs down the food. Woof!

Pluto sniffs the food. Woof!

Salem sniffs the food. Meow.

Nyan Cat wolfs down the food. Meow.

```
class TestAnimals {
    public static void main(String[] args) {
        Dog dug = new Dog("Dug", 5);
        Dog pluto = new Dog("Pluto", 84);
        Cat salem = new Cat("Salem", 18);
        Cat nyancat = new Cat("Nyan Cat", 3);
        Animal[] animals = {dug, pluto, salem, nyancat};
        feed(animals);
    }
    public static void feed(Animal[] animals) {
        for (Animal a : animals) {
            if (a.getAge() < 10) {
                System.out.print(a.getName() + " wolfs down the food. ");
            } else {
                System.out.print(a.getName() + " sniffs the food. ");
            }
            a.makeSound();
        }
    }
}
```

Discussion Question: Should `Animal` be an interface or an abstract class?

## 2 Inheritance

Cross out any lines that cause compile-time, and put an X through runtime errors (if any). What does the main program (in class D) output after removing these lines?

```
1 class A {
2     int x = 5;
3     public void m1 () {System.out.println("Am1-> " + x);}
4     public void m2 () {System.out.println("Am2-> " + this.x);}
5     public void update () {x = 99;}
6 }
7
8 class B extends A {
9     int x = 10;
10    public void m2 () {System.out.println("Bm2-> " + x);}
11    public void m3 () {System.out.println("Bm3-> " + super.x);}
12    public void m4 () {System.out.print("Bm4-> "); super.m2 ();}
13 }
14 class C extends B {
15     int y = x + 1;
16     public void m2 () {System.out.println("Cm2-> " + super.x);}
17     public void m3 () {System.out.println("Cm3-> " + super.super.x);}
18     public void m4 () {System.out.println("Cm4-> " + y);}
19     public void m5 () {System.out.println("Cm5-> " + super.y);}
20 }
21 class D {
22     public static void main (String[] args) {
23         B a0 = new A();
24         a0.m1 ();
25         A b0 = new B();
26         b0.m1 (); // class B hides a field in class A.
27         b0.m2 (); // you should never hide fields.
28         b0.m3 (); // as you'll see, it's confusing!
29         B b1 = new B();
30         b1.m3 ();
31         b1.m4 ();
32         A c0 = new C();
33         c0.m1 ();
34         C c1 = (A) new C();
35         A a1 = (A) c0;
36         C c2 = (C) a1;
37         c2.m4 ();
38         c2.m5 ();
39         ((C) c0).m3 (); // very tricky!
40         (C) c0.m3 ();
41         b0.update ();
42         b0.m1 ();
43         b0.m2 ();
44     }
45 }
```