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?

```java
class A {
    int x = 5;
    public void m1() {System.out.println("Am1-> " + x);}
    public void m2() {System.out.println("Am2-> " + this.x);} 
    public void update() {x = 99;}
}
class B extends A {
    int x = 10;
    public void m2() {System.out.println("Bm2-> " + x);} 
    public void m3() {System.out.println("Bm3-> " + super.x);} 
    public void m4() {System.out.println("Bm4-> "); super.m2();} 
}
class C extends B {
    int y = x + 1;
    public void m2() {System.out.println("Cm2-> " + super.x);} 
    public void m3() {System.out.println("Cm3-> " + super.super.x);} 
    public void m4() {System.out.println("Cm4-> " + y);} 
    public void m5() {System.out.println("Cm5-> " + super.y);} 
}
class D {
    public static void main (String[] args) {
        B a0 = new A();
        a0.m1();
        A b0 = new B();
        b0.m1(); // class B hides a field in class A.
        b0.m2(); // you should never hide fields.
        b0.m3(); // as you’ll see, it’s confusing!
        B b1 = new B();
        b1.m3();
        b1.m4();
        A c0 = new C();
        c0.m1();
        C c1 = (A) new C();
        A a1 = (A) c0;
        C c2 = (C) a1;
        c2.m4();
        c2.m5();
        ((C) c0).m3(); // very tricky!
        (C) c0.m3();
        b0.update();
        b0.m1();
        b0.m2();
    }
}
```