1 Creating Cats

Given the Animal class, fill in the definition of the Cat class so that when greet() is called, the label "Cat" (instead of "Animal") is printed to the screen. Assume that a Cat will make a "Meow!" noise, and that this is all caps for cats younger than 5 years old.

```java
public class Animal {
  protected String name, noise;
  protected int age;

  public Animal(String name, int age) {
    this.name = name;
    this.age = age;
    this.noise = "Huh?";
  }

  public String makeNoise() {
    if (age < 5) {
      return noise.toUpperCase();
    } else {
      return noise;
    }
  }

  public void greet() {
    System.out.println("Animal " + name + " says: " + makeNoise());
  }
}

public class Cat extends Animal {
}
```
2 Raining Cats and Dogs

Assume that Animal and Cat are defined as above. What will be printed at each of the indicated lines?

```java
public class TestAnimals {
    public static void main(String[] args) {
        Animal a = new Animal("Pluto", 10);
        Cat c = new Cat("Garfield", 6);
        Dog d = new Dog("Fido", 4);

        a.greet(); // (A) ______________________
        c.greet(); // (B) ______________________
        d.greet(); // (C) ______________________

        a = c;
        ((Cat) a).greet(); // (D) ______________________
        a.greet(); // (E) ______________________
    }
}

public class Dog extends Animal {
    public Dog(String name, int age) {
        super(name, age);
        noise = "Woof!";
    }

    @Override
    public void greet() {
        System.out.println("Dog " + name + " says: " + makeNoise());
    }

    public void playFetch() {
        System.out.println("Fetch, " + name + "!");
    }
}
```

Consider what would happen if we added the following to the bottom of main:

```java
a = new Dog("Spot", 10);
d = a;
```

Why would this code produce a compiler error? How could we fix this error?
3 An Exercise in Inheritance Misery

Cross out any lines that cause compile-time errors, and put an X through runtime errors (if any). What does the main program (in class D) output after removing these lines? Note: There are many cases covered here and possibly not enough time to finish in discussion. Remember that solutions will be posted online later this week.

```java
class A {
    public int x = 5;
    public void m1() {System.out.println("A m1-> " + x);}  // Cross out
    public void m2() {System.out.println("A m2-> " + this.x);} // Cross out
    public void update() {x = 99;}
}

class B extends A {
    public void m2() {System.out.println("B m2-> " + x);}  // Cross out
    public void m2(int y) {System.out.println("B m2y-> " + y);}  // Cross out
    public void m3() {System.out.println("B m3-> " + "called");}
}

class C extends B {
    public int y = x + 1;
    public void m2() {System.out.println("C m2-> " + super.x);}  // Cross out
    public void m4() {System.out.println("C m4-> " + super.super.x);} // Cross out
    public void m5() {System.out.println("C m5-> " + y);}  // Cross out
}

class D {
    public static void main (String[] args) {
        B a0 = new A();
        a0.m1();
        a0.m2(16);
        A b0 = new B();
        System.out.println(b0.x);
        b0.m1();
        b0.m2();
        b0.m2(61);
        B b1 = new B();
        b1.m2(61);
        b1.m3();
        A c0 = new C();
        c0.m2();
        C c1 = (A) new C();
        A a1 = (A) c0;
        C c2 = (C) a1;
        c2.m3();
        c2.m4();
        c2.m5();
        ((C) c0).m3();
        (C) c0.m3();
        b0.update();
        b0.m1();
    }
}
```