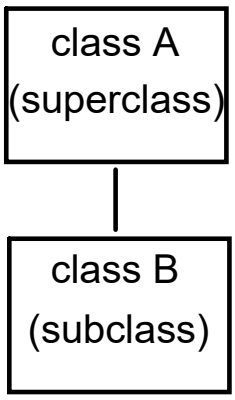
AP Computer Science A
Unit 7: Remaining Topics

Day 4:
Inheritance

Apr 19-1:12 PM

Inheritance (Parent/Child Classes)



important keywords for parent/child classes:

extends

links child class to its parent

class can only extend one class *but* can have several classes extending each other in a chain

`Book` → `ChildrensBook` → `BoardBook`

super

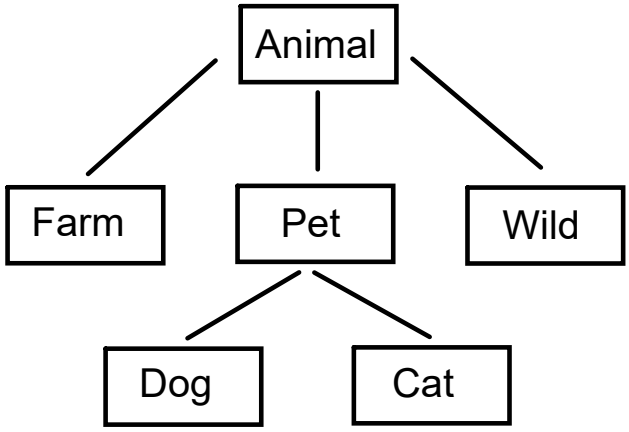
calls methods/constructors from parent class

if the parent class does not have the method, Java looks at the next parent class up the ladder

at the top ALL classes inherit from Object

note: class templates must be in the same folder and the parent must be compiled before child or you will get a compilation error

Aug 31-11:14 AM



Feb 19-6:55 AM

```
public class Animal {
   private int age;
   private double weight;
   public Animal(int a, double w) {
      if (a >= 0)
          age = a;
      else
          age = 0;
      if (w > 0)
          weight = w;
      else
          weight = -1;  //signals weight unknown
   }
   public String toString() {
      return "age: " + age + "\nweight: " + weight;
   }
  public int getAge() {
      return age;
   }
   public double getWeight() {
      return weight;
   }
}
```

Jan 30-9:02 AM

Notes:

- In a child class's CONSTRUCTOR, the call to the parent's CONSTRUCTOR must be the first line of code
- Even if you DO NOT code a call to the parent's constructor, Java DOES implicitly include one - it would be to the default constructor - if the parent does NOT have a default contructor, the child would not compile

```
public class Pet extends Animal{
    private String name;
    public Pet(String n, int a, double w){
        super(a, w);
        name = n;
    }
    public String toString() {
        return name + "\n" + super.toString();
    }
    public String getName() {
        return name;
    }
}
```

Jan 30-9:02 AM

```
public class Dog extends Pet {
   private String breed;
   public Dog(String n, String b, int a, double w){
       super(n, a, w);
       breed = b;
   }
   public String toString() {
       return super.toString() + "\nbreed: " + breed;
   }
   public void speak() {
       System.out.println("Woof");
   }
}
```

Mar 2-9:28 AM

```
public class Cat extends Pet {
   private boolean longhair;
   public Cat(String n, int a, double w,boolean l){
      super(n, a, w);
      longhair = l;
   }
   public String toString() {
     if (longhair)
      return super.toString() + "\nis a longhair cat";
     else
      return super.toString() + "\nis not a longhair cat";
   }
   public void speak() {
      System.out.println("Meow");
   }
}
```

Mar 2-9:28 AM

creating objects that extend

child types can be put into parent variables BUT NOT vice versa

Which lines will NOT compile and why?

```
Animal a1 = new Animal();
Animal a2 = new Animal(5, 15.2);
Animal a3 = new Animal(-2, 17.8);
Pet p1 = new Animal(3, 17.5);
Pet p2 = new Pet(4, 35.2);
Pet p3 = new Pet("Dolly", 6, 25.6);
Animal a4 = new Pet("Rascal",3, 17.5);
Dog d1 = new Animal(7, 16.8);
Dog d2 = new Pet("Fido", 8, 28.2);
Dog d3 = new Dog("Rex", 4, 15.4);
Dog d4 = new Dog("Samson", "Dalmation", 1, 40);
Cat c1 = new Cat("Sebastion", 3, 15, false);
Animal a5 = new Dog("Sandy","Beagle", 10, 45.2);
Pet p4 = new Dog("Tiger", "Greyhound", 2, 16.1);
Cat c2 = new Cat("Beatrice", 5, 17.2, true);
a5.speak();
d4.speak();
c2.speak();
p3.speak();
int age = c1.getAge();
```

Jan 30-9:05 AM

What prints out for each of the following?

```
System.out.println(a2 + "\n");
```

```
age: 5
weight: 15.2
```

```
System.out.println(a3 + "\n");
```

```
age: 0
weight: 17.8
```

```
System.out.println(p3 + "\n");
```

```
Dolly
age: 6
weight: 25.6
```

```
System.out.println(a4 + "\n");
```

```
Rascal
age: 3
weight: 17.5
```

```
System.out.println(d4 + "\n");
```

```
Samson
age: 1
weight: 40.0
breed: Dalmation
```

```
System.out.println(a5 + "\n");
```

```
Sandy
age: 10
weight: 45.2
breed: Beagle
```

```
System.out.println(c1 + "\n");
```

```
Sebastian
age: 3
weight: 15.0
is not a longhair cat
```

Jan 30-9:05 AM

Mar 10-6:51 PM