

# Objects, Inheritance and Polymorphism

Sidd and James

# Objectives

- Brief intro on objects and classes
- Polymorphism behaviors
- Python Implementation

# Objects

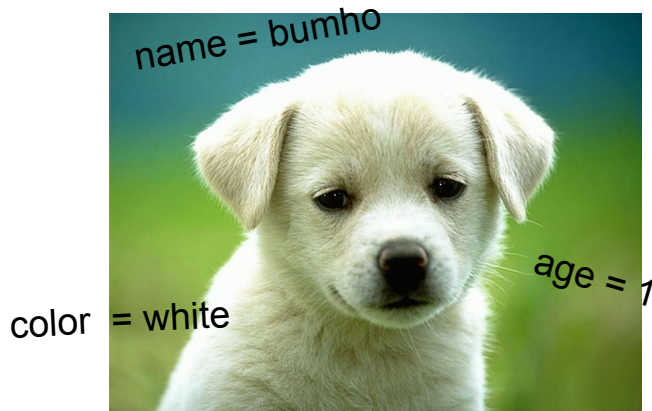
# Objects

- Object oriented programming



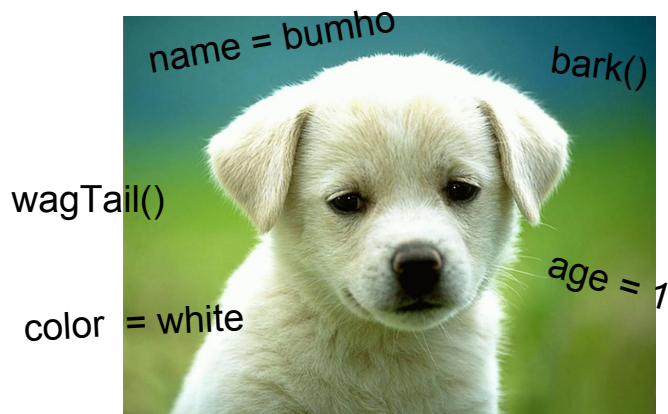
# Objects

- Object oriented programming
- States



# Objects

- Object oriented programming
- States
- Behaviors



# Classes vs Objects

# Classes vs Objects

Classes = templates





# Classes vs Objects

Classes = templates



Objects = actual instances



# Polymorphism

# Not Inheritance

## Tiger

age  
name  
color = orange  
numLegs = 4

makeSound() = "roar"  
moveForward(step) = x+=step  
attack() = ...

## Cat

age  
name  
color  
numLegs = 4  
domesticated

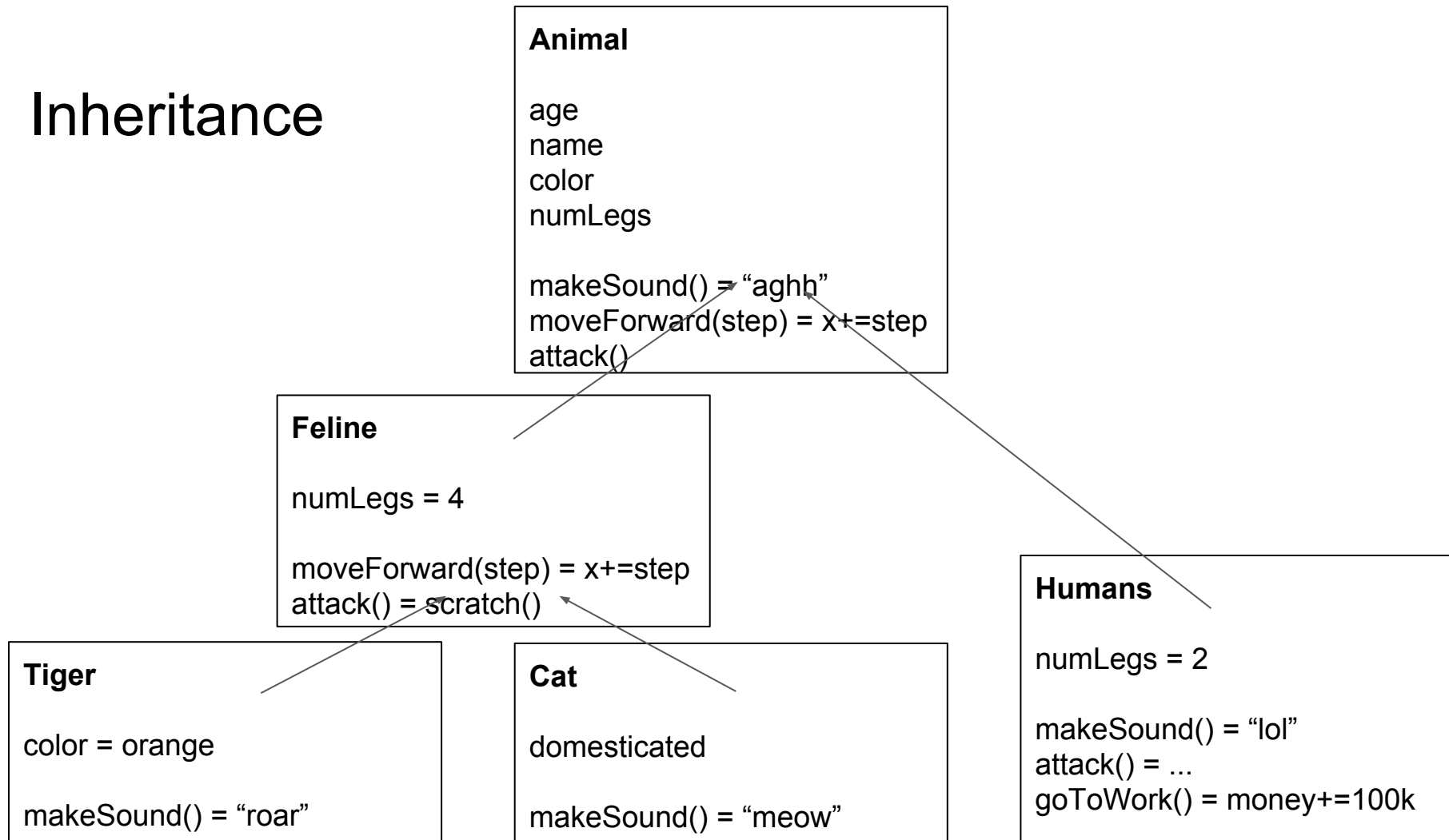
makeSound() = "meow"  
moveForward(step) = x+=step  
attack() = ...

## Humans

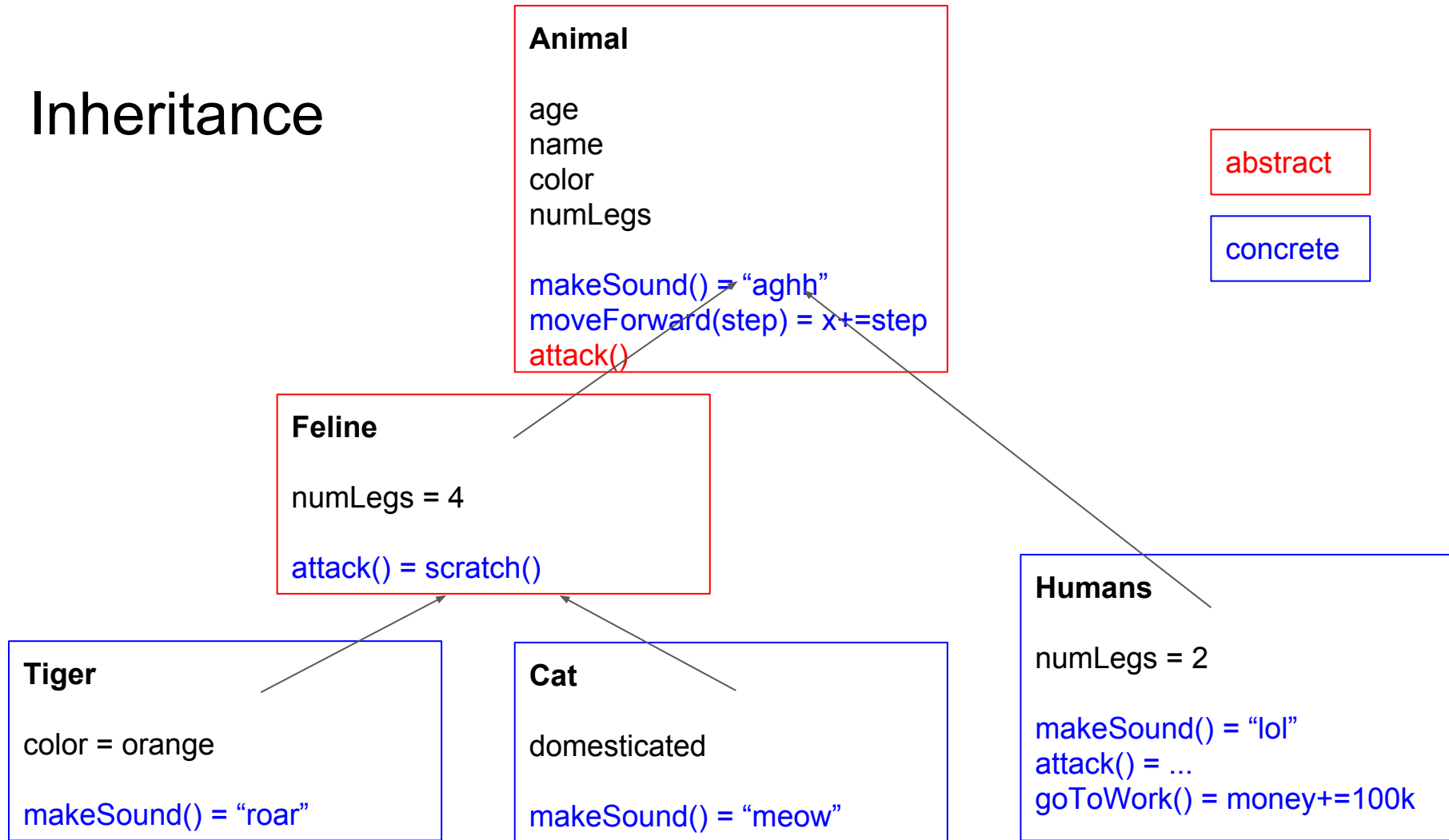
age  
name  
color  
numLegs = 2

makeSound() = "lol"  
moveForward(step) = x+=step  
attack() = ...  
goToWork() = money+=100k

# Inheritance



# Inheritance



# Polymorphism behaviors

```
obj Cat kitty = new Cat("Nippers" 5 "black")
```

# Polymorphism behaviors

```
obj Cat kitty = new Cat("Nippers" 5 "black")
```

```
(with kitty (makeSound) )
```


```
(with kitty (isDomesticated) )
```

# Polymorphism behaviors

```
obj Cat kitty = new Cat("Nippers" 5 "black")
```



Cat reference

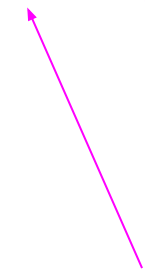


Cat object

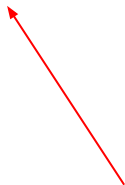


# Polymorphism behaviors

```
obj Feline kitty = new Cat("Nippers" 5 "black")
```



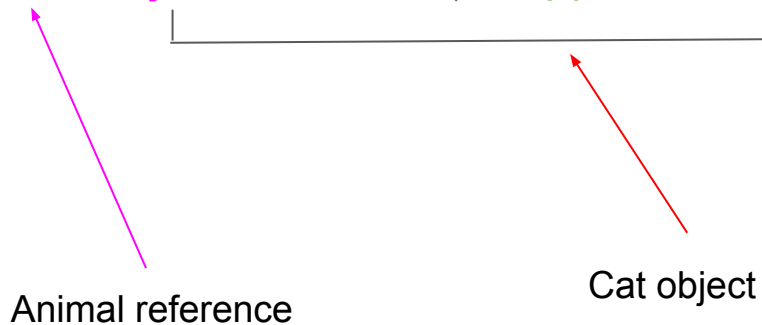
Feline reference



Cat object

# Polymorphism behaviors

```
obj Animal kitty = new Cat("Nippers" 5 "black")
```



# Polymorphism behaviors

```
obj Human kitty = new Cat("Nippers" 5 "black")
```

Human reference

Cat object

# Polymorphism behaviors

```
obj Cat kitty = new Animal("Nippers" 5 "black")
```

Cat reference

Animal object

# Polymorphism behaviors

```
obj Cat kitty = new Cat("Nippers" 5 "black")
```

```
(with kitty (makeSound) )
```

```
(with kitty (isDomesticated) )
```

```
(with kitty (goToWork) )
```

# Polymorphism behaviors

```
obj Cat kitty = new Cat("Nippers" 5 "black")
```

```
(with kitty (makeSound) ) => "meow"
```

```
(with kitty (isDomesticated) )
```

```
(with kitty (goToWork) )
```

# Polymorphism behaviors

```
obj Animal kitty = new Cat("Nippers" 5 "black")
```

```
(with kitty (makeSound) )
```

```
(with kitty (isDomesticated) )
```

```
(with kitty (goToWork) )
```

# Polymorphism behaviors

```
obj Animal kitty = new Cat("Nippers" 5 "black")
```

```
(with kitty (makeSound) ) => "meow"
```

```
(with kitty (isDomesticated) )
```

```
(with kitty (goToWork) )
```



# Where are we starting from?

- Homework 6
- Get natural syntax parsing out of the way
- Focus on OOP

# Template

- `class(ClassName (superclass) (parameter) functions`
- Resolves extending, overriding, inheritance

# Object Class

- obj `Animal kitty = new Cat("Nippers" 5 "black")`
- Pulls everything from Template
- Checks constructor and properly polymorphism

# Stuff we aren't doing

- Multiclass subclassing
- Type checking on parameters
- Interfaces
- Overloaded methods
- Generics
- Casting
- Better syntax

DEMO

# By the time we turn in code

- Abstract-ness
- Make structure more elegant
  - Make templates part of env
  - Decouple objects and declarations
  - Wrap all reference/object pairs in a VObjectBinding

# ~~Stuff we aren't doing~~ Future work

- Multiclass subclassing
- Type checking on parameters
- Interfaces
- Overloaded methods
- Generics
- Casting
- Better syntax

# Questions