Inheritance: Subclass

```
class Cat(Animal):
    def speak(self):
        print(“miao")
```

Add new methods via speak function

- add new function with `speak()`
  - Objects of Class `Cat` can be called with new methods
  - Objects of `Animal` throws error if called with `Cat`’s new method

```
x=Cat(3)
x.set_name("Jonnie")
x.speak()
print(x)
y=Animal(3)
y.speak()
```

Inheritance: Subclass

```
class Cat(Animal):
    def speak(self):
        print("miao")
    def __str__(self):
        return "cat:"+str(self.name)+":"+str(self.age)
```

Add new functions via speak method

Overrides__str__

- Subclass can override the **methods inherited from** superclass, then objects of subclass call the overrided methods

Inheritance: Subclass

- Subclass can have **methods with same name** as superclass
- For an instance of a class, look for a method name in **current class definition**
- If not found, look for method name **up the hierarchy** (in parent, then grandparent, and so on)

Inheritance: Subclass

```
class Cat(Animal):
    def speak(self):
        print("meow")
    def __str__(self):
        return "cat:"+str(self.name)+":"+str(self.age)
```

Add new functions via speak method

Overrides__str__

```
x=Cat(3)
x.set_name("Jonnie")
x.speak()
print(x)
```

```
miao
cat:Jonnie:3
```