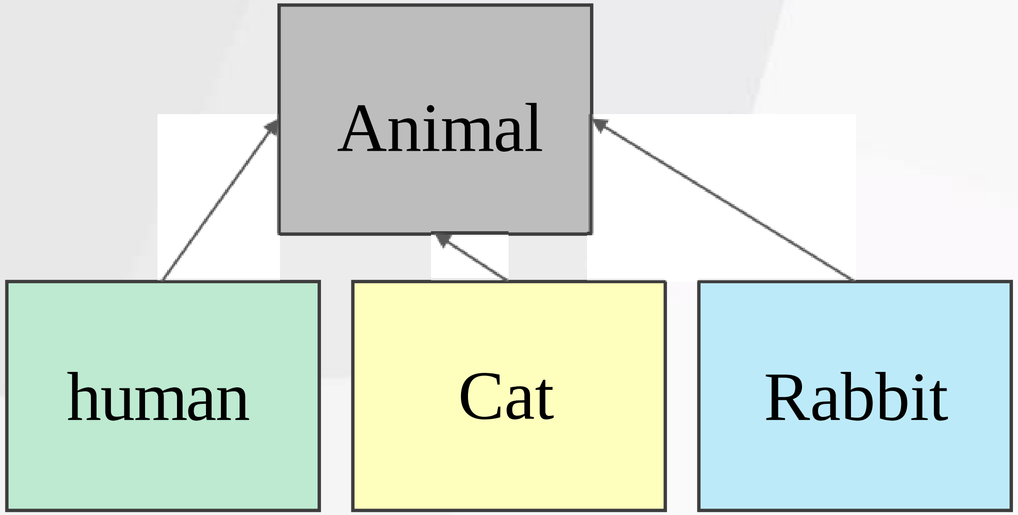
Hierarchies

- **parent class** (superclass)
- **child class** (subclass)
  - **inherits** all attributes and behaviors of parent class
  - **add** more **info/attributes**
  - **add** more **behavior/methods**
  - **override** behavior



Inheritance: Parent Class

```
class Animal:
    def __init__(self, age):
        self.age = age
        self.name = None
    def get_age(self):
        return self.age
    def get_name(self):
        return self.name
    def set_age(self, newage):
        self.age = newage
    def set_name(self, newname):
        self.name = newname
    def __str__(self):
        return "animal:"+str(self.name)+":"+str(self.age)
```

Inheritance: Subclass

```
class Cat(Animal):
```

Cat inherits all attributes of Animal:

Attributes: age, name

Method:

__int__()

get_age(),

get_name()

set_age(),

set_name()

__str__()

Inheritance: Subclass

```
class Cat(Animal):
      def __init__(self, age, color):
          super().__init__(age)
          color=color
```

its parent →

Call the methods from its parent

```
x=Cat(3,”brown”)
print(x.age, x.color)
```

```
3 brown
```