Lecture-8

Inheritance in Python

Content

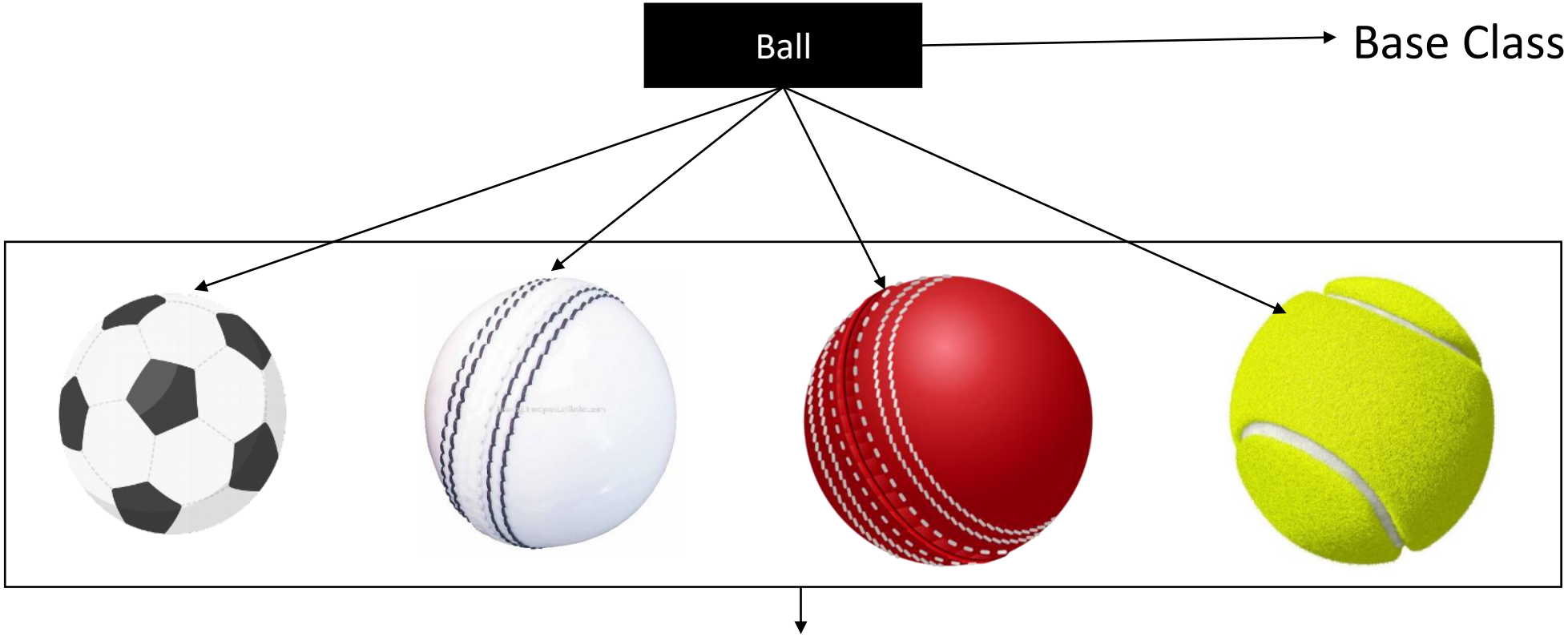
- Defines Inheritance with Example
- Use of super() Function
- Overriding in Python

What is Inheritance?

Inheritance allows us to define a class that inherits all the methods and properties from another class.

- **Parent class:** is the class being inherited from, also called base class.
- **Child class** is the class that inherits from another class, also called derived class.

Inheritance Example



Create Base Class

```
#create base calss
class Ball():
    def __init__(self):
        print(f'Ball created!')

    def who_am_i(self):
        print(f'i am ball')
```

Create Child Class

```
#create a Child(football) class

class Football(Ball):
  def __init__(self):
    #create instances of Ball
    Ball.__init__(self)
    print(f'football created!')
```

Create Object & Call Base Class Method

```
#create object
myFootball = Football()
```

```
#call base method
myFootball.who_am_i()
```

Benefits of Inheritance

- It represents real-world relationships well.
- It provides **reusability** of a code. We don't have to write the same code again and again. Also, it allows us to add more features to a class without modifying it

Use the Super() Function

- Python has a **super()** function that will make the child class inherit all the methods and properties from its parent
- By using the super() function, you do not have to use the name of the parent element, it will automatically inherit the methods and properties from its parent.

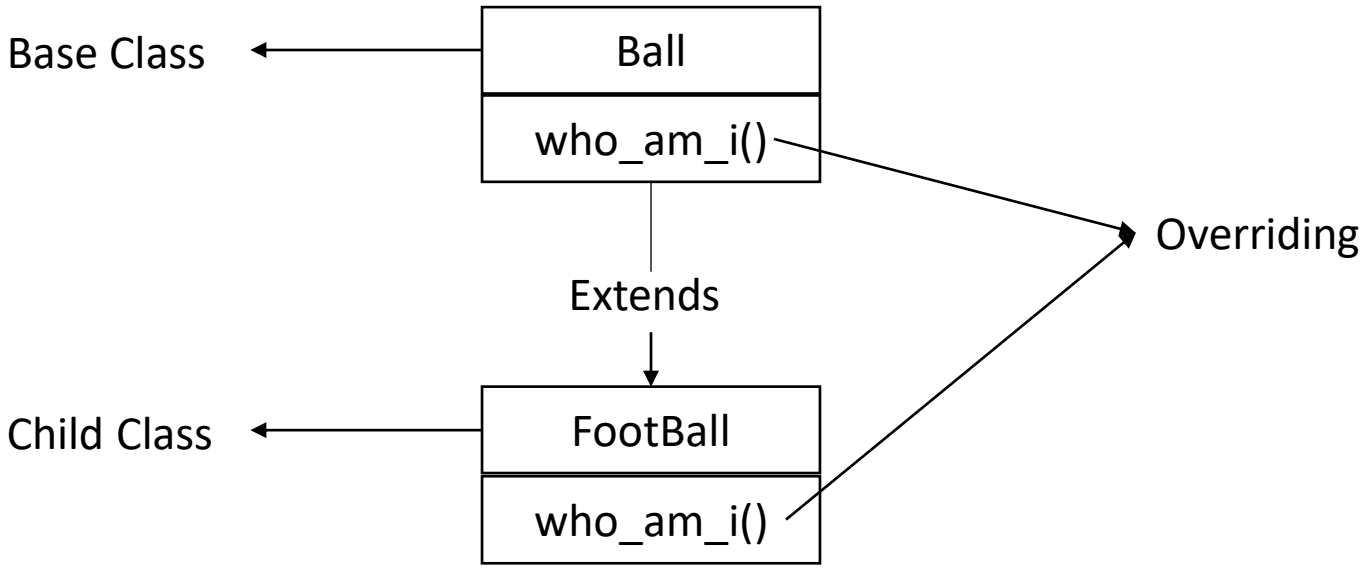
Use the Super() Function

```
class Football(Ball):
    def __init__(self):
        #create instances of Ball
        super().__init__(self)
        print(f'football created!')
```

Overriding

When a method in a subclass has the same name, same parameters or signature and same return type (or sub-type) as a method in its super-class, then the method in the subclass is said to **override** the method in the super-class.

Overriding Example



Overriding Example

```
#overriding
class Football(Ball):
  def __init__(self):
    #create instances of Ball
    Ball.__init__(self)
    print(f'football created!')
  #overide base class method
  def who_am_i(self):
    print(f'i am football!')
```

Thank You