

1:33 AM 5.2KB/s 83

Cut Paste Format Painter New Reset Layout Section Font Paragraph Drawing Slides Outline

170 We cannot use super() to access parent class instance variables from child class. We should use self only.

171 From child class constructor and instance methods, we can call parent class constructors , instance methods , class methods and static methods by using super().

172 From child class , class method we cannot access parent class constructor and instance methods directly by using super(). But we can access parent class static and class methods.  
Reason: Class method no way related to object. Without object also we can call class method. But constructor and instance methods are always associated with object.

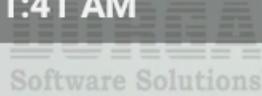
173 From child class static method, we cannot use super() to call parent class members. But indirectly we can call parent class static and class methods.

# From child class constructor and instance methods, we can call parent class constructors , instance methods , class methods and static methods by using super().

From child class , class method we cannot access parent class constructor and instance methods directly by using super(). But we can access parent class static and class methods.

Reason: Class method no way related to object. Without object also we can call class method. But constructor and instance methods are always associated with object.

1:41 AM



583KB/s 82

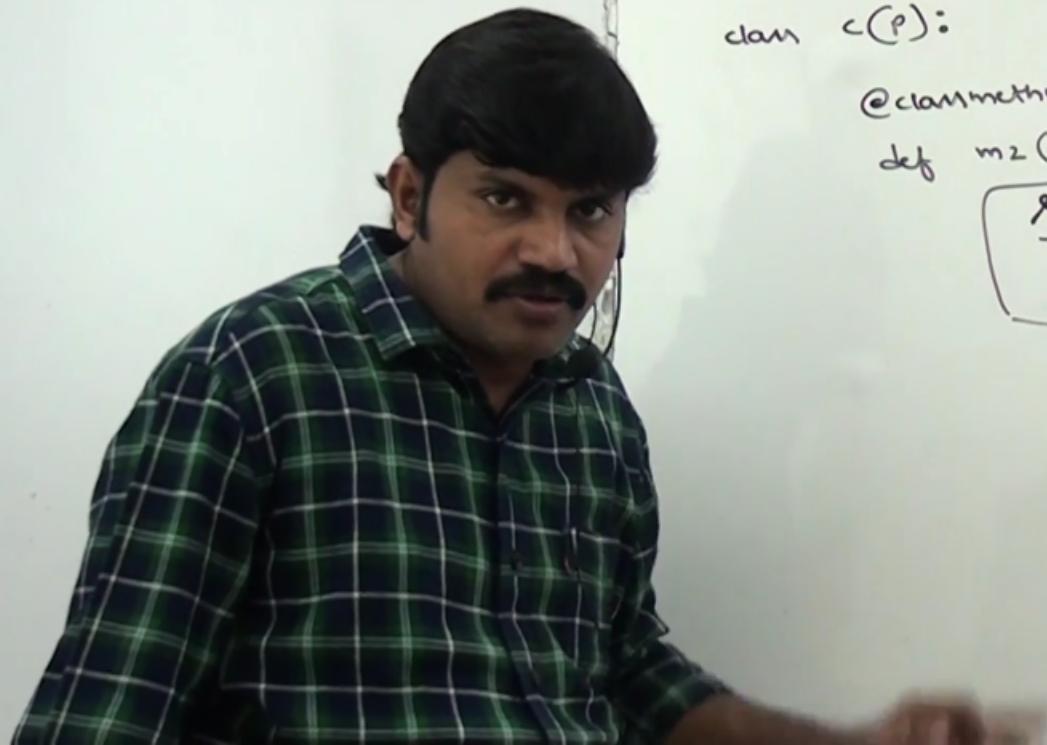
From child class static method, we cannot use super() to call parent class members. But indirectly we can call parent class static and class methods.



**From class method of child class, how to call parent class constructor and instance methods indirectly???**

```
class P:  
    def __init__(self):  
        print("Parent constructor")  
  
    def m1(self):  
        print("Parent Instance method")
```

```
class C(P):  
    @classmethod  
    def m2(cls):  
        Super(C, cls).__init__(cls)  
        Super(C, cls).m1(cls)
```



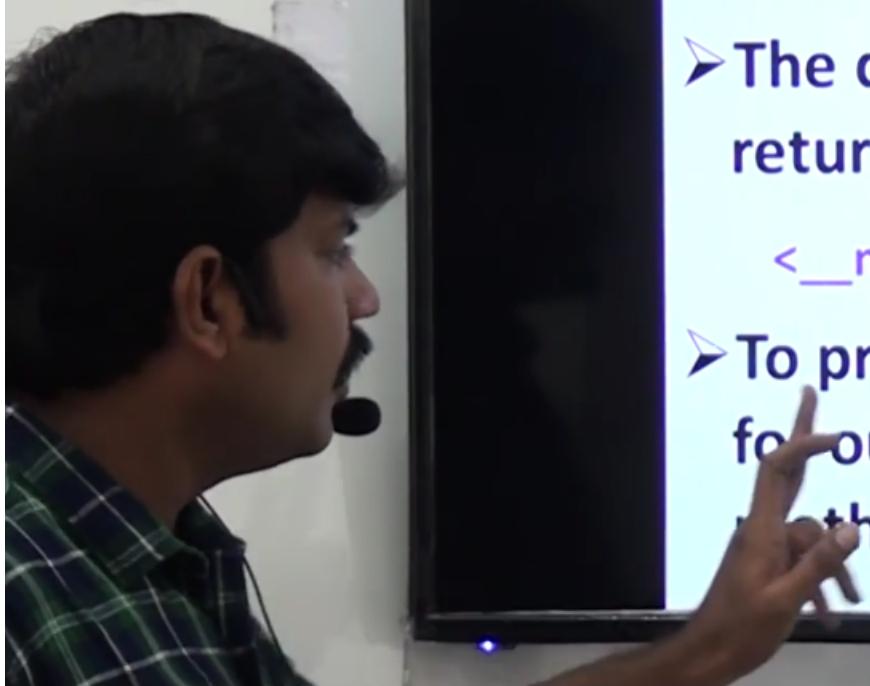


How to call Parent class static  
and class methods from child  
class static method ???

```
class P:  
    @classmethod  
    def m2(c1):  
        print(' Parent class method')  
  
    @staticmethod  
    def m3():  
        print(' Parent static method')  
  
class C(P):  
    @staticmethod  
    def m2():  
        super(C, C).m2()  
    super(C, C).m3()
```



```
1 class P:  
2     @classmethod  
3         def m2(cls):  
4             print('Parent Class Method')  
5     @staticmethod  
6     def m3():  
7         print('Parent static method')  
8  
9  
10 class C(P):  
11     @staticmethod  
12     def m2():  
13         super(C,C).m2()  
14         super(C,C).m3()  
15  
16 C.m2()
```



## Importance of `__str__()` method:

- Whenever we are trying to print any object reference, internally `__str__()` method will be called.
- The default implementation of this method returns the string in the following format:  
`<__main__.Student object at 0x0000000000280D748>`
- To provide meaningful string representation for our object, we have to override `__str__()` method in our class.