Inheritance

Thursday, July 17, 2025 7:17 PM

It is the phenomenon of deriving the property of one class to another class.

- The class from which the property is derived is called as parent or base or super class.
- The class to which the property is derived is called as child or derived or sub class.

Types of Inheritance:

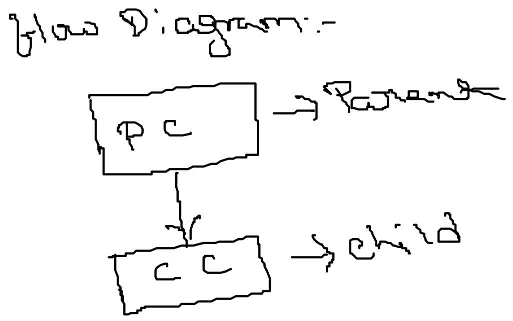
Inheritance is classified into 5 types:

1. Single Inheritance
2. Multi-level inheritance
3. Multiple Inheritance
4. Hierarchial Inheritance
5. Hybrid Inheritance

1. Single Inheritance: It is the type of inheritance where the property of one parent class gets inherited by one child class.

Syntax:

```
class PC:
        S B
Class CC(PC):
        S B
```



Eg:

```
class A:
   a=10
   b=20
   @classmethod
   def disp(cls):
      print(cls.a,cls.b)
ob=A()
class B(A):
   c=30
   def __init__(self,m):
      self.m=m
ob1=B(300)
ob.disp()
ob1.disp()
```

Constructor Chaining: It is a process of calling or revoking the parent class constructor inside the child class.

Syntax:

i. Super().__init__(args)
ii. Super(child,self).__init__(args)
iii. Pname.__init__(self,args)

Eg:

```
class Stud:
   def __init__(self,name,roll):
      self.name=name
      self.roll=roll

   def disp(self):
      print(self.name,self.roll)

class Marks(Stud):
    def __init__(self,name,roll,mark):
        super().__init__(name,roll)
        self.mark=mark

    def dis(self):
        print(self.mark)
s1=Marks('Aman',111,87)
s1.disp()
s1.dis()
```

Method Chaining: It is a phenomenon of calling the parent class method in the child class.

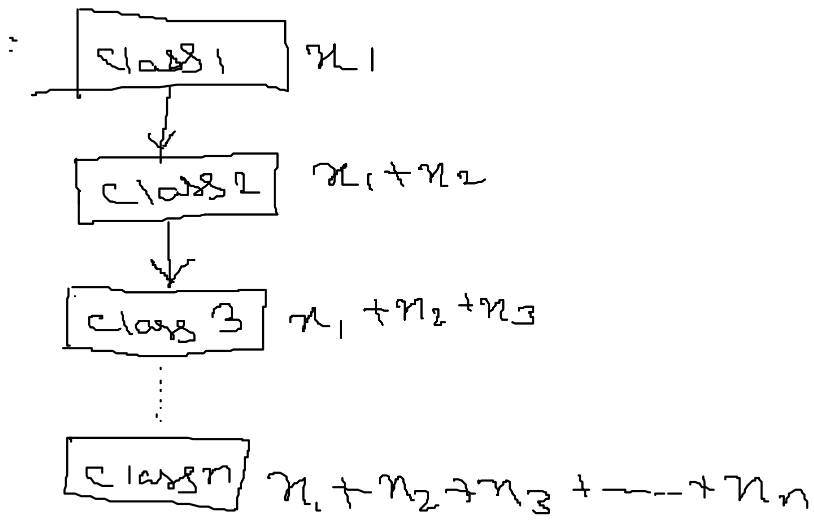
Syntax:

i. Super().mname(args)
ii. Super(child,self/cls).mname(args)
iii. Pname.mname(self/cls,args)

Eg:

```
class Person:
    def s_name(self,name):
        self.name=name
        return self
    def greet(self):
        print(f'Hello, My name is {self.name}')
        return self
class Student(Person):
    def s_sub(self,subject):
        self.subject=subject
        return self
    def s_show(self):
        print(f'I study {self.subject}')
s=Student()
s.s_name('Aman').greet().s_sub('Math').s_show()
```

2. Multi-level Inheritance: Deriving the property multiple times is called multi-level inheritance. Here 1 child class acts as the parent class for other child class.

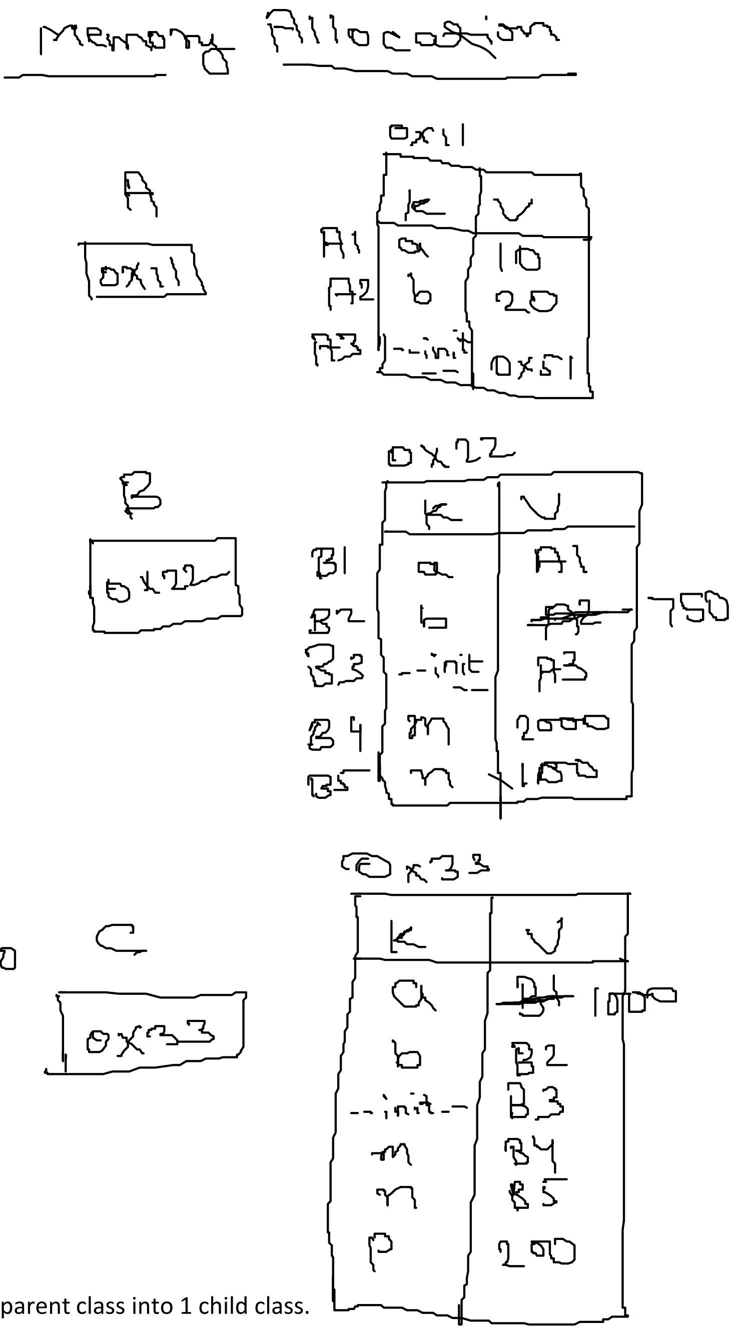


Syntax:

```
Class C1:
        S B
Class C2(C1):
        SB
Class C3(C2):
        S B
        .
        .
        .
Class Cn(Cn-1):
        S B
```

Eg:

```
class A:
   a=10
   b=20
   def __init__(self,c,d):
      self.c=c
      self.d=d
class B(A):
   m=2000
   n=100
   b=750
class C(B):
   a=1000
   p=200
print(A.a,A.b)
print(B.a,B.b,B.m,B.n)
print(C.a,C.b,C.m,C.n,C.p)
```



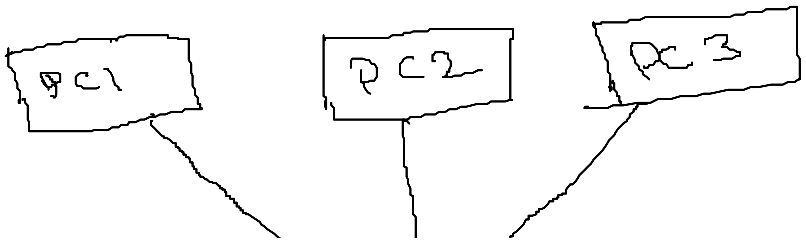
O/P :-

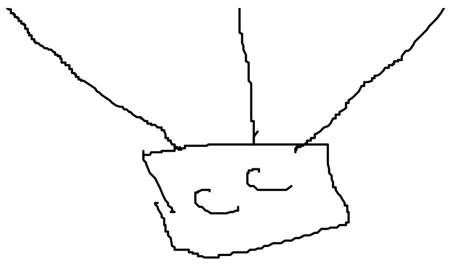
```
10   20

10   750   2000   100

1000   750   2000   100   200
```

3. Multiple Inheritance: Inheriting the property of multiple parent class into 1 child class.



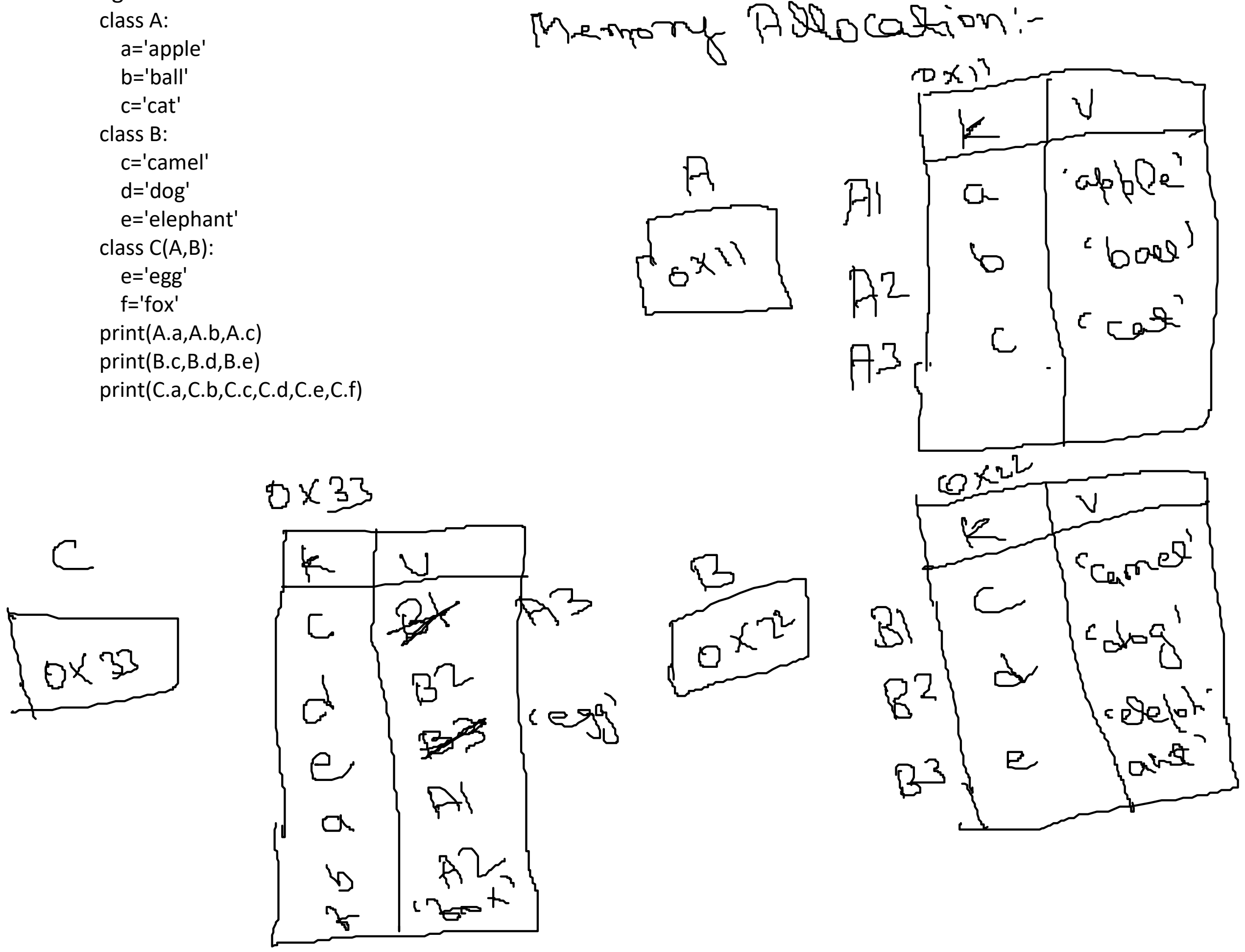


Syntax:

```
Class PC1:
        S B
Class PC2:
        S B
Class PC3:
        S B
        .
        .
        .
Class PCn:
        S B
Class CC(PC1,PC2,PC3........PCn): #here inherting the property starts from the last PC  and then goes to first PC in reverse.
        S B
```

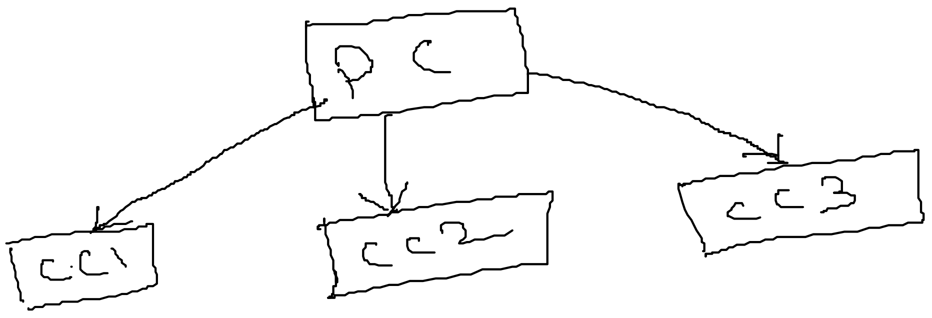
Eg:

```
class A:
   a='apple'
   b='ball'
   c='cat'
class B:
   c='camel'
   d='dog'
   e='elephant'
class C(A,B):
   e='egg'
   f='fox'
print(A.a,A.b,A.c)
print(B.c,B.d,B.e)
print(C.a,C.b,C.c,C.d,C.e,C.f)
```



4. Hierarchical Inheritance: The property of 1 parent class is inherited by multiple child class.

4. Hierarchical Inheritance: The property of 1 parent class is inherited by multiple child class.



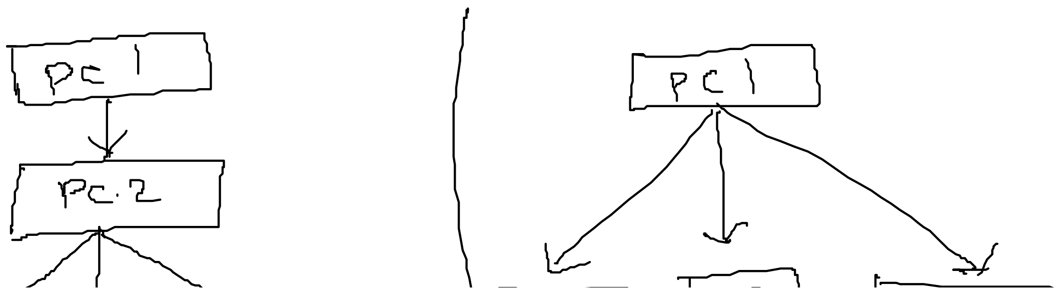
Syntax:

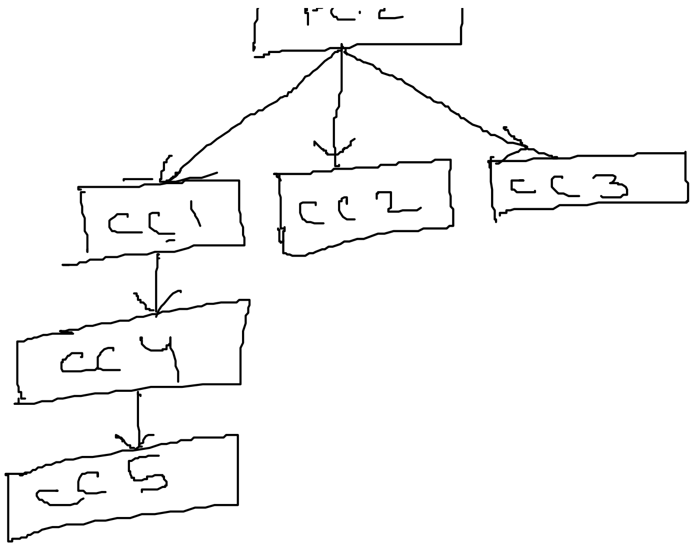
```
Class PC:
        S B
Class CC1(PC):
         S B
Class CC2(PC):
        S B
        .
        .
        .
Class CCn(PC):
        S B
```

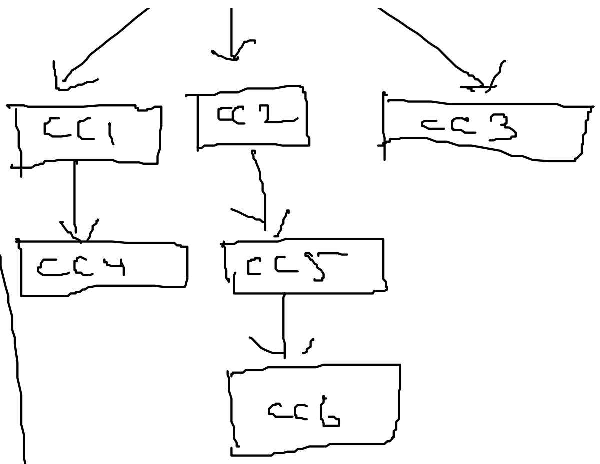
Eg:

```
class A:
   a=10
   b=20
   def __init__(self,c,d):
      self.c=c
      self.d=d
class B(A):
   a=1000
   @classmethod
   def disp(cls):
      print(cls.a)
class C(A):
   m=500
   @staticmethod
   def sam():
      print('hii')
print(A.a,A.b)
print(B.a,B.b)
print(C.a,C.b,C.m)
B.disp()
C.sam()
```

5. Hybrid Inheritance: It is a combination of all the 4 types of inheritance




CC1
CC2
CC3
CC4
CC5


CC1
CC2
CC3
CC4
CC5
CC6