

EXPERIMENT NO : 3B

Python Programs To Implement Different Types of Inheritance.

NAME : AKASH RAMKRIT YADAV

ID.NO: VU4F2122016

BATCH : A

BRANCH : IT

DIV : A

Aim : Python Programs to Implement different types of Inheritance.

THEORY:

OUTPUT:

Python 3.11.0a4 (main, FEB 27 2023, 12:57:32) [MSC v.1929 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

#AKASH YADAV ID.NO:VU4F2122016 EXP:3B DATE:27/2/2023

Types of Inheritance in Python

Types of Inheritance depend upon the number of child and parent classes involved.

There are five types of inheritance in Python

>> 1 Single Inheritance:

2 Multiple Inheritance:

3 Multilevel Inheritance :

4 Hierarchical Inheritance:

5 Hybrid Inheritance:

#1 Single Inheritance:

Single inheritance enables a derived class to inherit properties from a single parent class, thus enabling code reusability and the addition of new features to existing code.

```
#Base class
class india:
    def f1(self):
        print("This function is in parent class.")
```

```
# Derived class
class mumbai(india):
    def f2(self):
        print("This function is in child class.")
```

```
# Driver's code
obj=mumbai()
obj.f1()
obj.f2()
```

```
>>>This function is in parent class.
      This function is in child class.
```

#2 Multiple Inheritance:

When a class can be derived from more than one base class this type of inheritance is called multiple inheritances. In multiple inheritances, all the features of the base classes are inherited into the derived class.

```
# Base class1
class Father:
    Fathername=""
    def Father(self):
```

```
print(self.Fathername)
```

```
# Base class2
```

```
class Mother:
```

```
    Mothername=""
```

```
    def Mother(self):
```

```
        print(self.Mothername)
```

```
# Derived class
```

```
class Child(Father,Mother):
```

```
    def Parent(self):
```

```
        print("FatherName : ",self.Fathername)
```

```
        print("MotherName : ",self.Mothername)
```

```
# Driver's code
```

```
a1=Child()
```

```
a1.Fathername="AKASH YADAV"
```

```
a1.Mothername="AKANSHA YADAV"
```

```
a1.Parent()
```

```
>>>FatherName : AKASH YADAV
```

```
    MotherName : AKANSHA YADAV
```

#3 Multilevel Inheritance :

In multilevel inheritance, features of the base class and the derived class are further inherited into the new derived class.

This is similar to a relationship representing a child and a grandfather.

```
# Base class
```

```
class GrandMother:
```

```
    def __init__(self,GrandMothername):
```

```
        self.GrandMothername=GrandMothername
```

```
# Intermediate class
```

```
class Mother(GrandMother):
```

```

def __init__(self,Mothername,GrandMothername):
    self.Mothername=Mothername

    GrandMother.__init__(self,GrandMothername)

# Derived class
class Son(Mother):
    def __init__(self,Sonname,Mothername,GrandMothername):
        self.Sonname=Sonname
        Mother.__init__(self,Mothername,GrandMothername)

    def print_name(self):
        print('GrandMother name :', self.GrandMothername)
        print("Mother name :", self.Mothername)
        print("Son name :", self.Sonname)

# Driver code
akash = Son('Krishna', 'Devaki', 'Marisha')
print("LORD KRISHNA PARENTS : CODE BY AKASH YADAV")
akash.print_name()

>>>LORD KRISHNA PARENTS : CODE BY AKASH YADAV
    GrandMother name : Marisha
    Mother name : Devaki
    Son name : Krishna

```

#4 Hierarchical Inheritance:

When more than one derived class are created from a single base this type of inheritance is called hierarchical inheritance.

In this program, we have a parent (base) class and two child (derived) classes.

```

class Parent:
    def f1(self):
        print("This function is in parent class.")

```

```
class Child1(Parent):  
    def f2(self):  
        print("This function is in Child 1")
```

```
class Child2(Parent):  
    def f3(self):  
        print("This function is in Child 2")
```

```
akash1=Child1()  
akash2=Child2()  
akash1.f1()  
akash1.f2()  
akash2.f1()  
akash2.f3()
```

```
>>>This function is in parent class.  
This function is in Child 1  
This function is in parent class.  
This function is in Child 2
```

#5 Hybrid Inheritance:

Inheritance consisting of multiple types of inheritance is called hybrid inheritance

```
class PVPPCOE:  
    def f1(self):  
        print("This function is in class PVPPCOE ")
```

```
class IT(PVPPCOE):  
    def f2(self):  
        print("This function is in class IT ")
```

```
class CS(PVPPCOE):  
    def f3(self):  
        print("This function is in class CS ")
```

```
class ITSA(IT,PVPPCOE):  
    def f4(self):  
        print("This function is in class ITSA ")
```

Driver's code

AKASH=ITSA()

AKASH.f1()

AKASH.f2()

```
>>>This function is in class PVPPCOE  
This function is in class IT
```