# Single Inheritance  
class Base:  
 def \_\_init\_\_(self, name, id ) :  
 self.name = name  
 self.id = id  
  
 def show(self):  
 print("Name:", self.name)  
 print("ID:", self.id)  
  
  
class Derived(Base):  
 def \_\_init\_\_(self, name, id, salary):  
  
 self.salary = salary  
  
 Base.\_\_init\_\_(self, name, id)  
  
 def show1(self):  
 print("Salary:", self.salary)  
  
  
d = Derived("Jareena", 12, 35000)  
d.show()  
d.show1()

Output:

Name: Jareena

ID: 12

Salary: 35000

----------------------------------------------------------------------------------------------------------------------------------------

# Multilevel Inheritance Inheritance  
class Base:  
 def \_\_init\_\_(self, name, id ) :  
 self.name = name  
 self.id = id  
  
 def show(self):  
 print("Name:", self.name)  
 print("ID:", self.id)  
  
  
class Derived(Base):  
 def \_\_init\_\_(self, name, id, salary):  
  
 self.salary = salary  
  
 Base.\_\_init\_\_(self, name, id)  
  
 def show1(self):  
 print("Salary:", self.salary)  
  
  
class Derived1(Derived):  
 def \_\_init\_\_(self, name, id, salary, address):  
 self.address = address  
  
 Derived.\_\_init\_\_(self, name, id, salary)  
  
 def show2(self):  
 print("Address:", self.address)  
  
  
  
d = Derived1 ("Jareena", 12, 35000, 'Pune')  
d . show ()  
d . show1 ()  
d . show2 ()

Output:

Name: Jareena

ID: 12

Salary: 35000

Address: Pune

------------------------------------------------------------------------------------------------------------------------------------------

# User of super()  
class Base:  
 def \_\_init\_\_(self, name, id ) :  
 self.name = name  
 self.id = id  
  
 def show(self):  
 print("Name:", self.name)  
 print("ID:", self.id)  
  
  
class Derived(Base):  
 def \_\_init\_\_(self, name, id):  
 self.sname = name  
 self.sid = id  
  
 super().\_\_init\_\_("Jareena", 13)  
  
 def showdata(self):  
 print("Name:", self.sname)  
 print("ID:", self.sid)  
  
  
d = Derived("Batul", 10)  
d.show()  
d.showdata()

**Output:**

Name: Jareena

ID: 13

Name: Batul

ID: 10

-----------------------------------------------------------------------------------------------------------------------------------------

#super demo  
# parent class  
class Person():  
 def \_\_init\_\_(self, name, age):  
 self.name = name  
 self.age = age  
  
 def display(self):  
 print(self.name, self.age)  
  
# child class  
class Student(Person):  
 def \_\_init\_\_(self, name, age, dob):  
 self.sName = name  
 self.sAge = age  
 self.dob = dob  
 # inheriting the properties of parent class  
 super().\_\_init\_\_("Rahul", age)  
  
 def displayInfo(self):  
 print(self.sName, self.sAge, self.dob)  
  
obj = Student("Mayank", 23, "16-03-2000")  
obj.display()  
obj.displayInfo()

Output:

Rahul 23

Mayank 23 16-03-2000

------------------------------------------------------------------------------------------------------------------------------------------

# Python program to demonstrate  
# multiple inheritance  
  
# Base class1  
class Mother:  
 mothername = ""  
  
 def mother(self):  
 print(self.mothername)  
  
# Base class2  
  
  
class Father:  
 fathername = ""  
  
 def father(self):  
 print(self.fathername)  
  
# Derived class  
  
  
class Son(Mother, Father):  
 def parents(self):  
 print("Father :", self.fathername)  
 print("Mother :", self.mothername)  
  
  
# Driver's code  
s1 = Son()  
s1.fathername = "RAM"  
s1.mothername = "SITA"  
s1.parents()

Output:

Father : Sadique

Mother : Jareena

------------------------------------------------------------------------------------------------------------------------------------------

# Python program to demonstrate  
# Hierarchical inheritance  
  
  
# Base class  
class Parent:  
 def func1(self):  
 print("This function is in parent class.")  
  
# Derived class1  
  
  
class Child1(Parent):  
 def func2(self):  
 print("This function is in child 1.")  
  
# Derivied class2  
  
  
class Child2(Parent):  
 def func3(self):  
 print("This function is in child 2.")  
  
  
# Driver's code  
object1 = Child1()  
object2 = Child2()  
object1.func1()  
object1.func2()  
object2.func1()  
object2.func3()

Output:

This function is in parent class.

This function is in child 1.

This function is in parent class.

This function is in child 2.

------------------------------------------------------------------------------------------------------------------------------------------

# Python program to demonstrate  
# hybrid inheritance  
  
  
class School:  
 def func1(self):  
 print("This function is in school.")  
  
  
class Student1(School):  
 def func2(self):  
 print("This function is in student 1. ")  
  
  
class Student2(School):  
 def func3(self):  
 print("This function is in student 2.")  
  
  
class Student3(Student1, School):  
 def func4(self):  
 print("This function is in student 3.")  
  
  
# Driver's code  
object = Student3()  
object.func1()  
object.func2()

Output:

This function is in school.

This function is in student 1.

------------------------------------------------------------------------------------------------------------------------------------------