**#Python program to different types of inheritance**

**Multilevel**

class A:

def first\_method(self):

print("Method of class A...")

class B(A):

def second\_method(self):

print("Method of class B....")

class C(B):

def third\_method(self):

print("Method of class C...")

if \_\_name\_\_=="\_\_main\_\_":

ob=C()

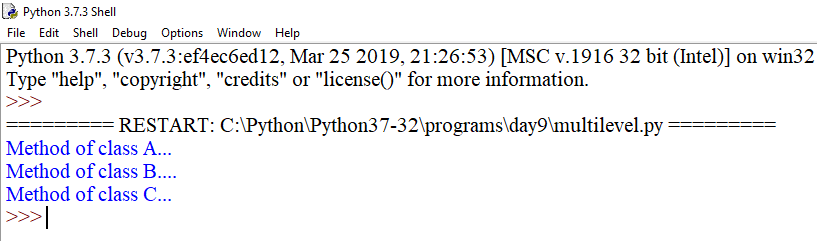
ob.first\_method()

ob.second\_method()

ob.third\_method()

#C().third\_method()

**OUTPUT**

****

**Multiple**

class A:

def first\_method(self):

print("Method of class A...")

class B:

def second\_method(self):

print("Method of class B....")

class C(A,B):

def third\_method(self):

print("Method of class C...")

if \_\_name\_\_=="\_\_main\_\_":

ob=C()

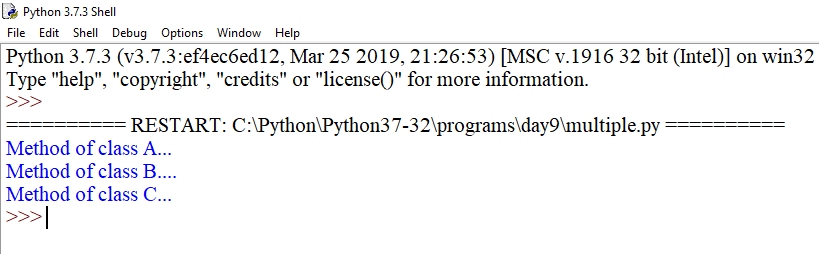
ob.first\_method()

ob.second\_method()

ob.third\_method()

#C().third\_method()

**OUTPUT**



class A:

def first\_method(self):

print("Method of class A...")

class B:

def first\_method(self):

print("Method of class B....")

class C(A,B):

def third\_method(self):

print("Method of class C...")

if \_\_name\_\_=="\_\_main\_\_":

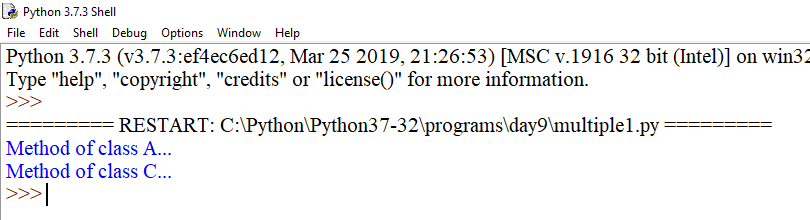
ob=C()

ob.first\_method()

ob.third\_method()

#C().third\_method()

**OUTPUT**

****

**Override**

class Wolf:

def \_\_init\_\_(self,name,color):

self.name=name

self.color=color

def bark(self):

print("Grr.........")

class Dog(Wolf):

def bark(self):

print("Woof")

pet1=Dog("Tommy","Brown")

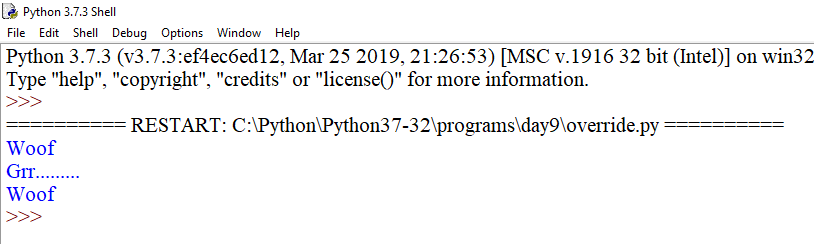
pet1.bark()

pet2=Wolf("Jimmy","Grey")

pet2.bark()

Dog("abc","xyz").bark()

**OUTPUT**

****