## Task 2 Question

class Airline\_travellers:

def \_\_init\_\_(self,foodservice="Normal",cost="37000/="):

self.foodservice=foodservice

self.cost=cost

def fuel1(self):

return("In domestic flight we have the {} food\_service and it cost is {}".format(self.foodservice,self.cost))

class Businessclass:

def fuel2(self,foodservice,cost,facility):

self.foodservice=foodservice

self.cost=cost

self.facility=facility

return("In business flight we have the {} food\_service and it cost is {} with fully {} ".format(self.foodservice,self.cost,self.facility))

class VIPclass(Airline\_travellers,Businessclass):

def fuel3(self,foodservice="luxiorous",cost="100,000/=",facility="A.C compartment"):

self.foodservice=foodservice

self.cost=cost

self.facility=facility

return("In VIPClass flight we have the {} food\_service and it cost is {} with fully {} ".format(self.foodservice,self.cost,self.facility))

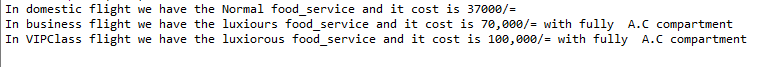
Al=VIPclass()

print(Al.fuel1())

print(Al.fuel2("luxiours","70,000/=","A.C compartment"))

print(Al.fuel3())

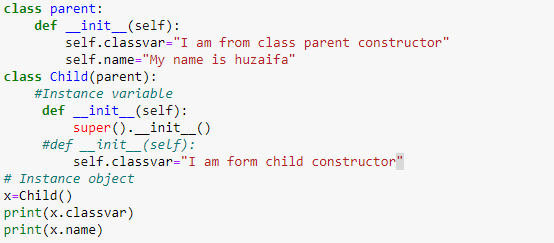
## Result



## Task 1 Question

# super() function in Python:

Python super function provides us the facility to refer to the parent class explicitly. It is basically useful where we have to call superclass functions. It returns the proxy object that allows us to refer parent class by ‘super’.



C:\Users\HASSAN ENTERPRISES\Desktop\sp Result.png

To understand Python super function we must know about the inheritance. In Python inheritance, the subclasses are inherited from the superclass.

## Inheritance And Its Types:

# Single Inheritance:

In single inheritance parentclass or BaseClass is inherit with single class childclass or subclass

Code reusability- we do not have to write the same code again and again, we can just inherit the properties we need in a child class.

# Multiple Inheritance:

In multiple inheritance one parent class is inherit with two child class Code reusability- we do not have to write the same code again and again, we can just inherit the properties we need in a multiple class.

## Task 3 Question

class Classic\_Umbrella:

def \_\_init\_\_(self,material,style,prints,size,used\_for):

self.material=material

self.style=style

self.prints=prints

self.size=size

self.used\_for=used\_for

def ForRain(self):

return("This is the most common type of modern foldable umbrellas. Commonly made with {} to keep the rain off, these umbrellas are your everyday portable shelters from storms.Its style is {} and its printing is{} Its size is {}.This Umbrella is used for Both {} \n".format(self.material,self.style,self.prints,self.size,self.used\_for))

class Bubble\_Umbrella:

def \_\_init\_\_(self,fmaterial,fstyle,fprints,fsize,fused\_for):

self.fmaterial=fmaterial

self.fstyle=fstyle

self.fprints=fprints

self.fsize=fsize

self.fused\_for=fused\_for

def ForSunprotection(self,colors):

self.fcolors=colors

return("This is the most commmon type of foldable umbrella.commonly made up of {} to Sunprotection and rainfalls.These are great for {} who love to peer out at the rain while keeping their faces dry.Its style is {} and its printing is{} Its size is {}.Its has diffrents colos {}".format(self.fmaterial,self.fused\_for,self.fstyle,self.fprints,self.fsize,self.fcolors))

class Automatic\_Umbrella(Classic\_Umbrella,Bubble\_Umbrella):

def \_\_init\_\_(self,material,style,prints,size,used\_for,fmaterial,fstyle,fprints,fsize,fused\_for):

Classic\_Umbrella.\_\_init\_\_(self,material,style,prints,size,used\_for)

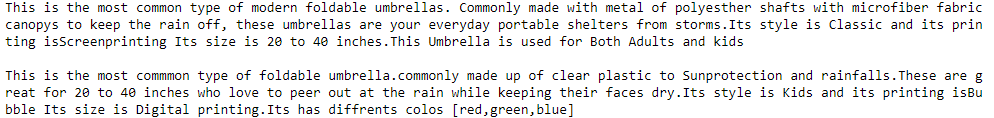
Bubble\_Umbrella.\_\_init\_\_(self,fmaterial,fstyle,fprints,fsize,fused\_for)

AU=Automatic\_Umbrella("metal of polyesther shafts with microfiber fabric canopys","Classic","Screenprinting","20 to 40 inches","Adults and kids","clear plastic","Kids","Bubble","Digital printing","20 to 40 inches")

print(AU.ForRain())

#AH=Automatic\_Umbrella("clear plastic","Kids","Bubble","Digital printing","20 to 40 inches")

print(AU.ForSunprotection("[red,green,blue]"))



# 