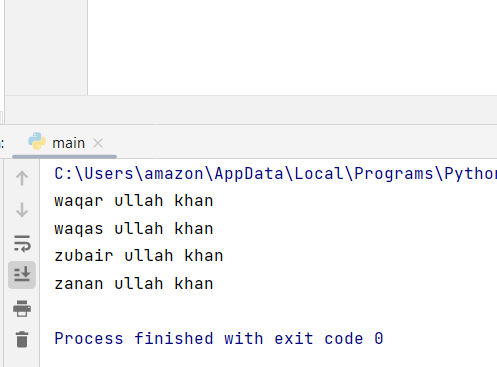
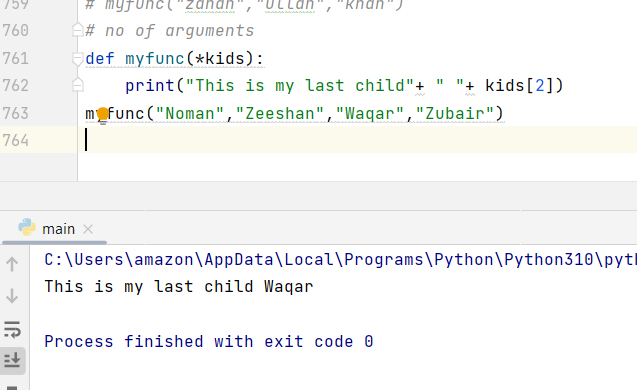
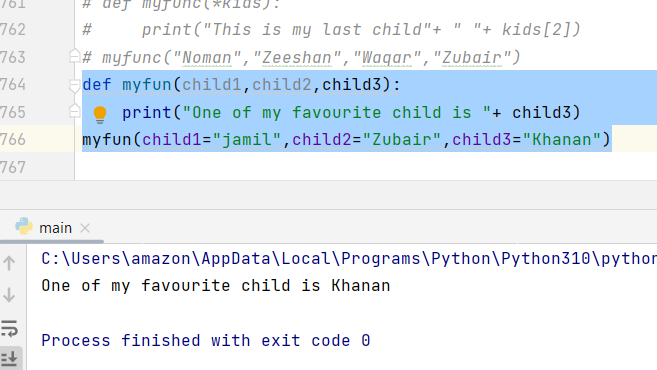
**Python Advanced 3:**

*# def myfunc():  
# print("This is my first functions ")  
# myfunc()  
# def myfunc(firstname):  
# print("first name"+" "+ firstname+" "+ "Refrences")  
# myfunc("Jamil")  
# myfunc("Haris")  
# myfunc("Waqar")*def myfunc(fname,mname,lname):  
 print(f"{fname} {mname} {lname}")  
myfunc("waqar","ullah","khan")  
myfunc("waqas","ullah","khan")  
myfunc("zubair","ullah","khan")  
myfunc("zanan","ullah","khan")

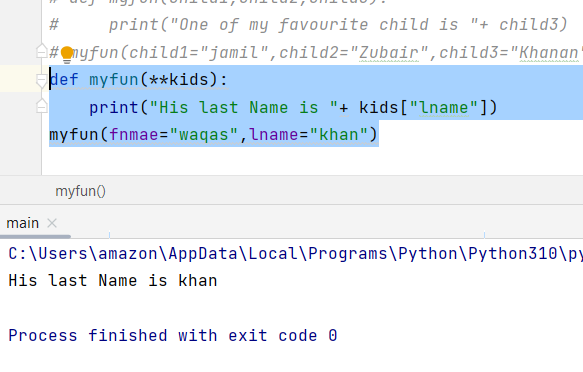


def myfunc(\*kids):  
 print("This is my last child"+ " "+ kids[2])  
myfunc("Noman","Zeeshan","Waqar","Zubair")

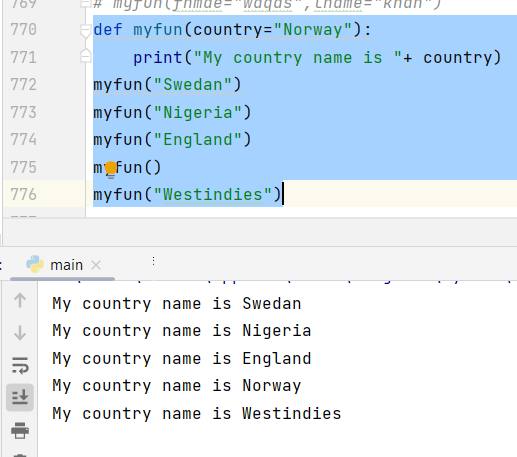
def myfun(child1,child2,child3):  
 print("One of my favourite child is "+ child3)  
myfun(child1="jamil",child2="Zubair",child3="Khanan")



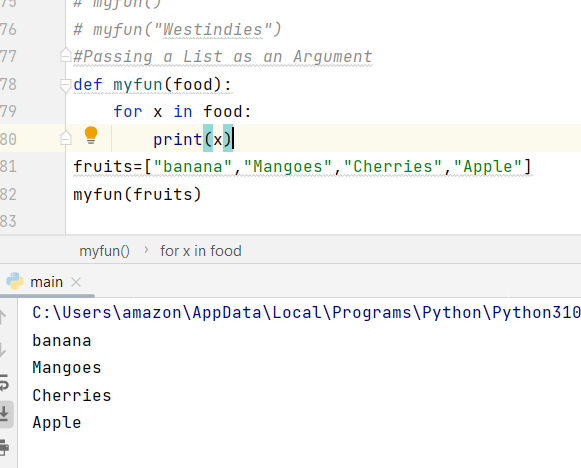
def myfun(\*\*kids):  
 print("His last Name is "+ kids["lname"])  
myfun(fnmae="waqas",lname="khan")



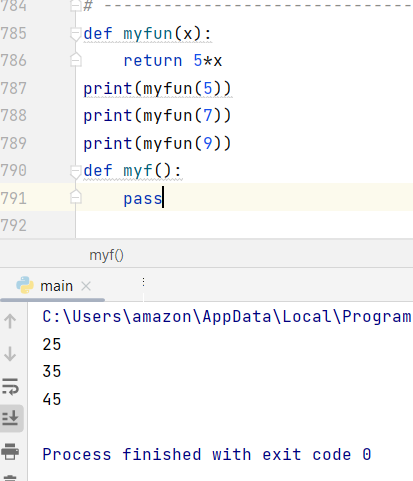
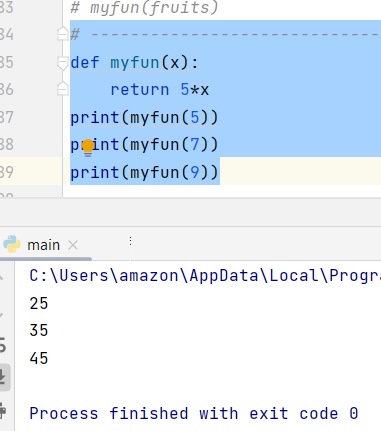
def myfun(country="Norway"):  
 print("My country name is "+ country)  
myfun("Swedan")  
myfun("Nigeria")  
myfun("England")  
myfun()  
myfun("Westindies")



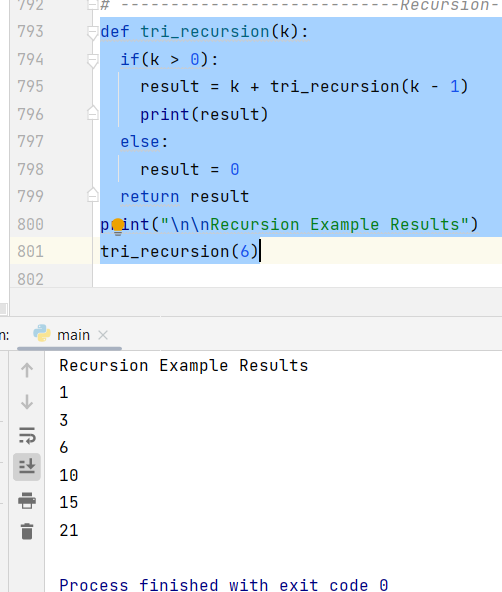
def myfun(food):  
 for x in food:  
 print(x)  
fruits=["banana","Mangoes","Cherries","Apple"]  
myfun(fruits)



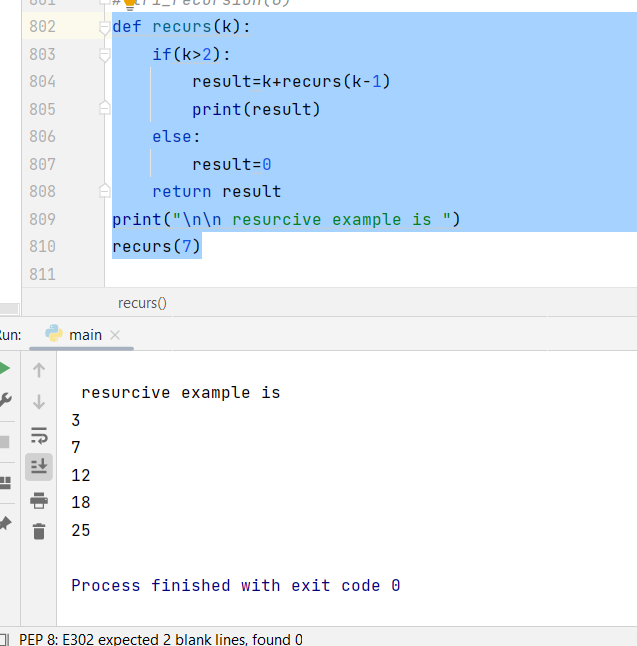
*# ------------------------------------------------Return values-----------------*def myfun(x):  
 return 5\*x  
print(myfun(5))  
print(myfun(7))  
print(myfun(9))

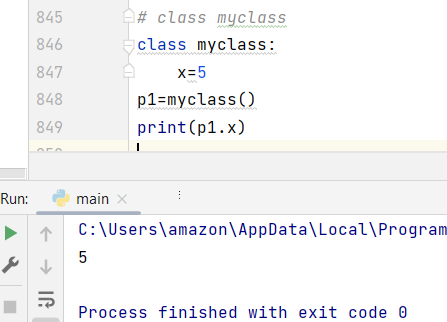
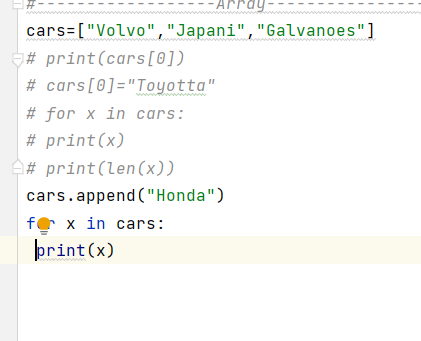
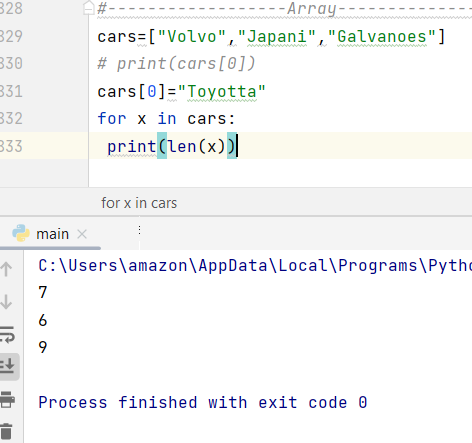
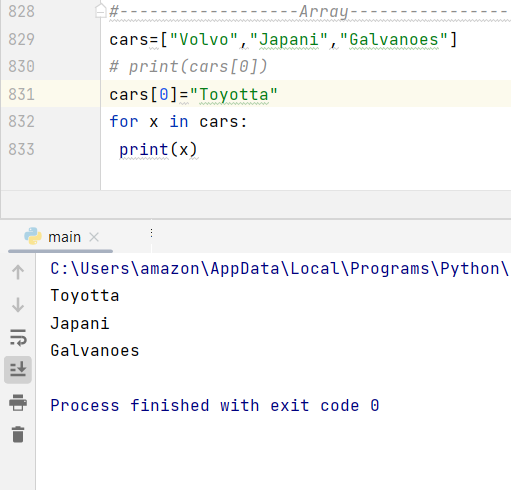
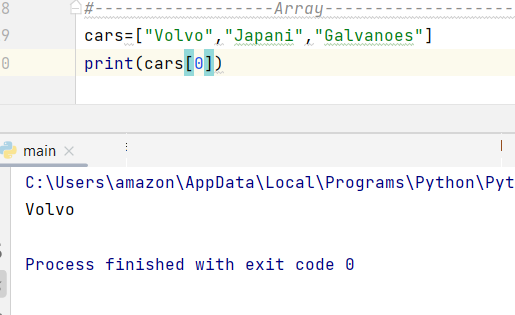
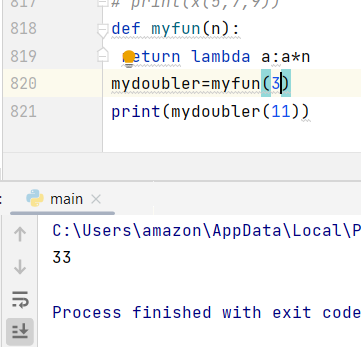
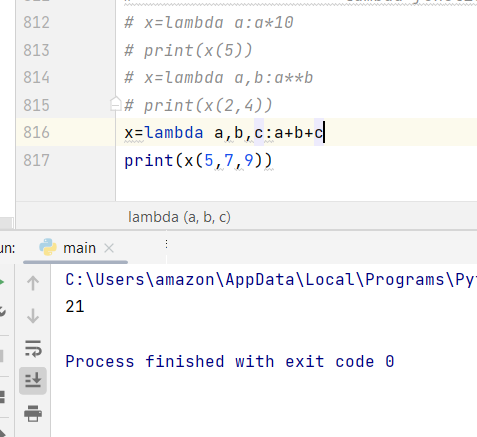
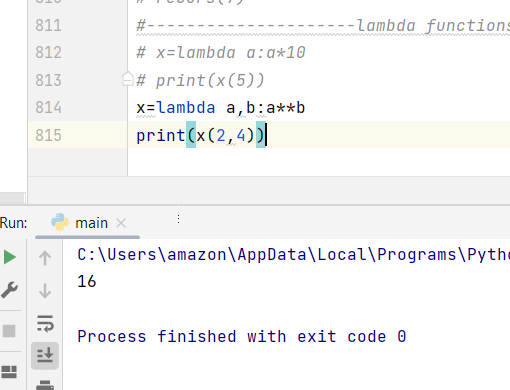
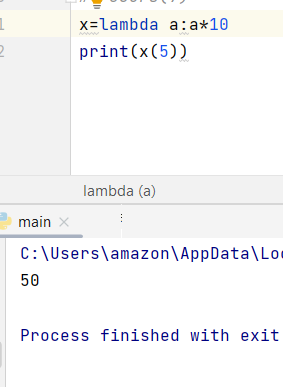


def tri\_recursion(k):  
 if(k > 0):  
 result = k + tri\_recursion(k - 1)  
 print(result)  
 else:  
 result = 0  
 return result  
print("\n\nRecursion Example Results")  
tri\_recursion(6)

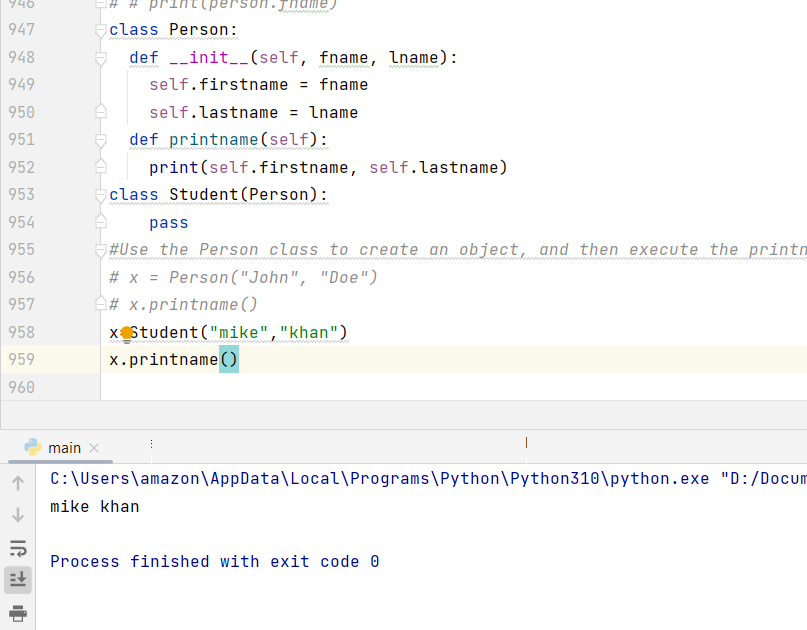
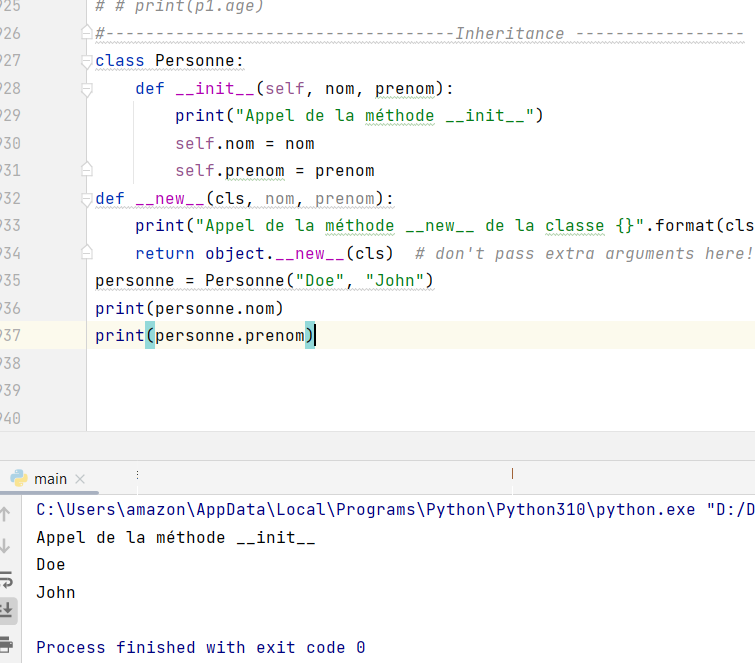
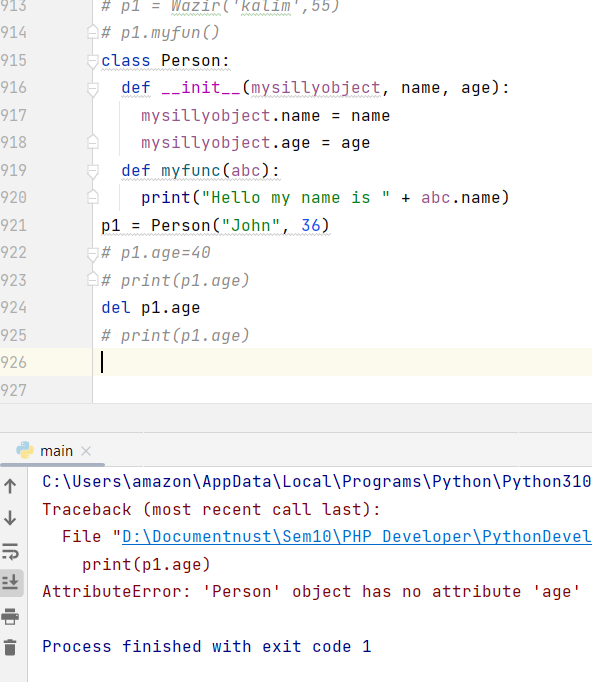
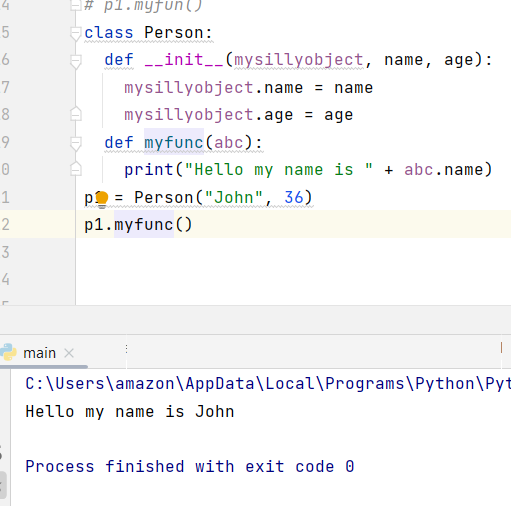
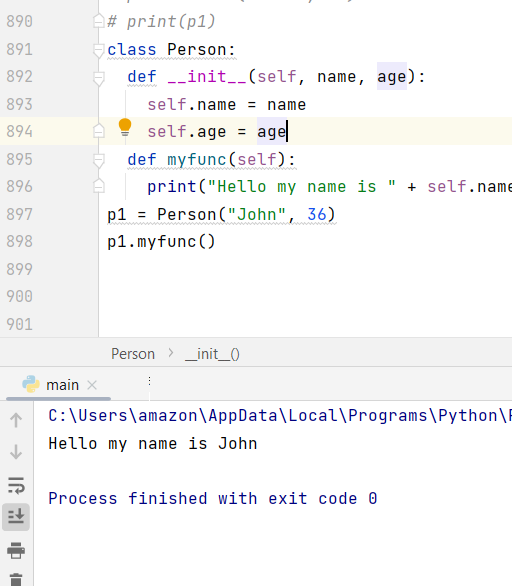
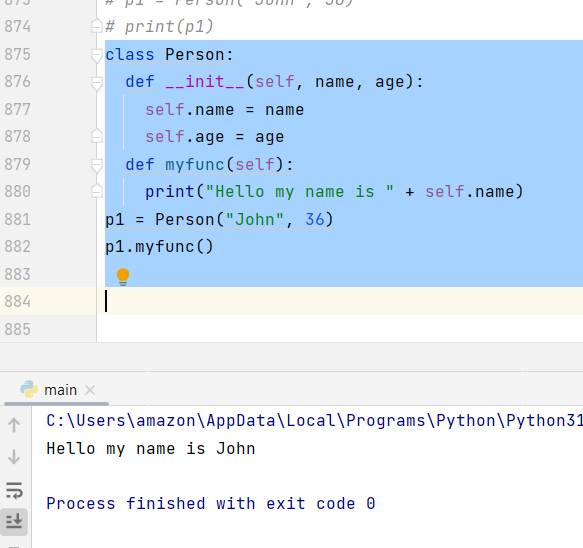
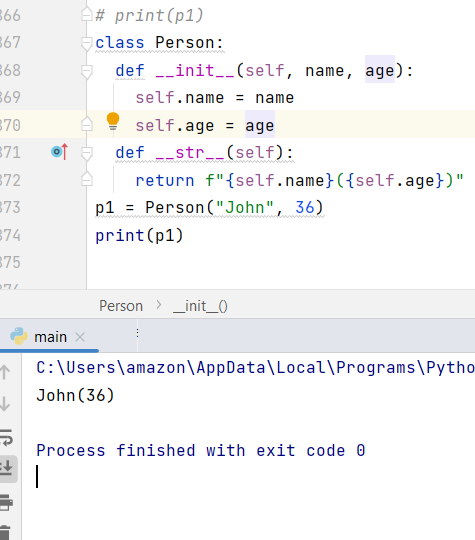
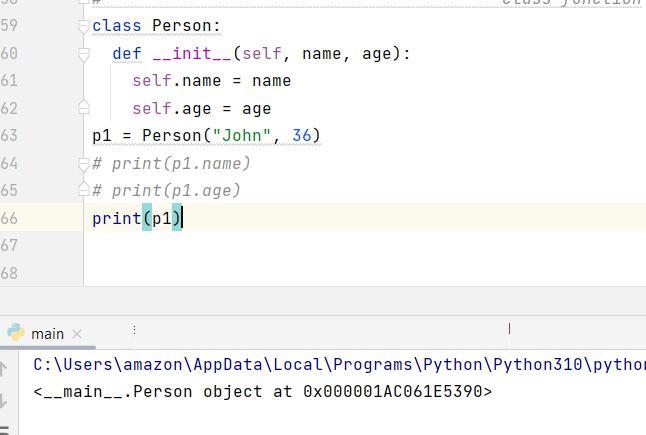
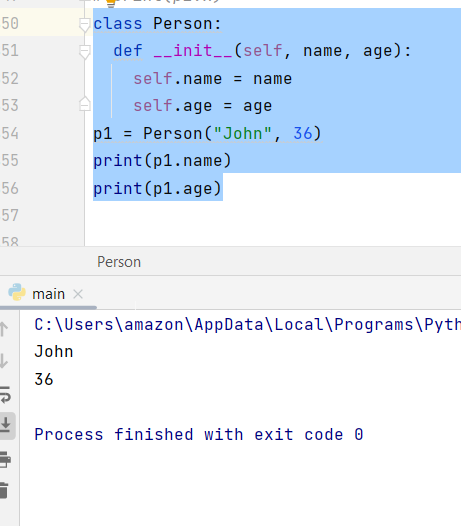


def recurs(k):  
 if(k>2):  
 result=k+recurs(k-1)  
 print(result)  
 else:  
 result=0  
 return result  
print("\n\n resurcive example is ")  
recurs(7)

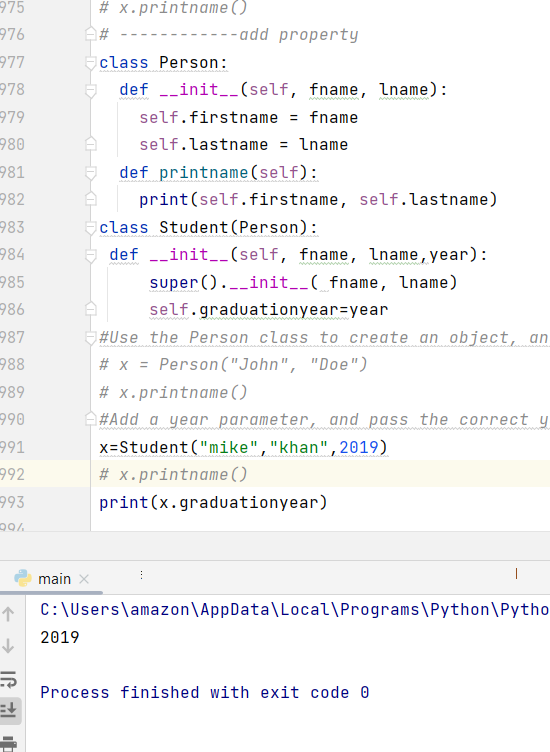
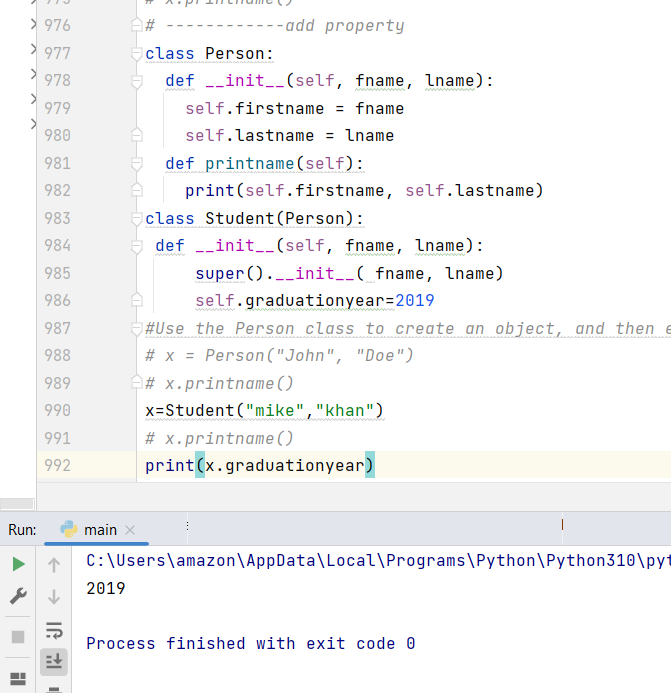
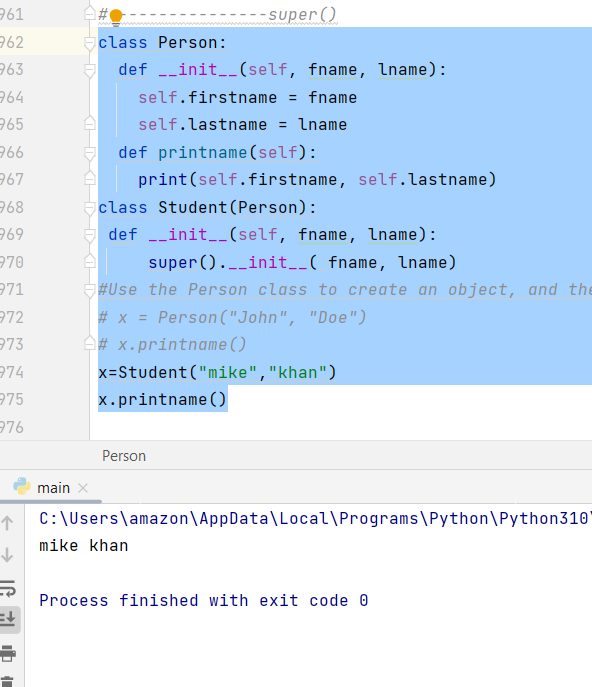




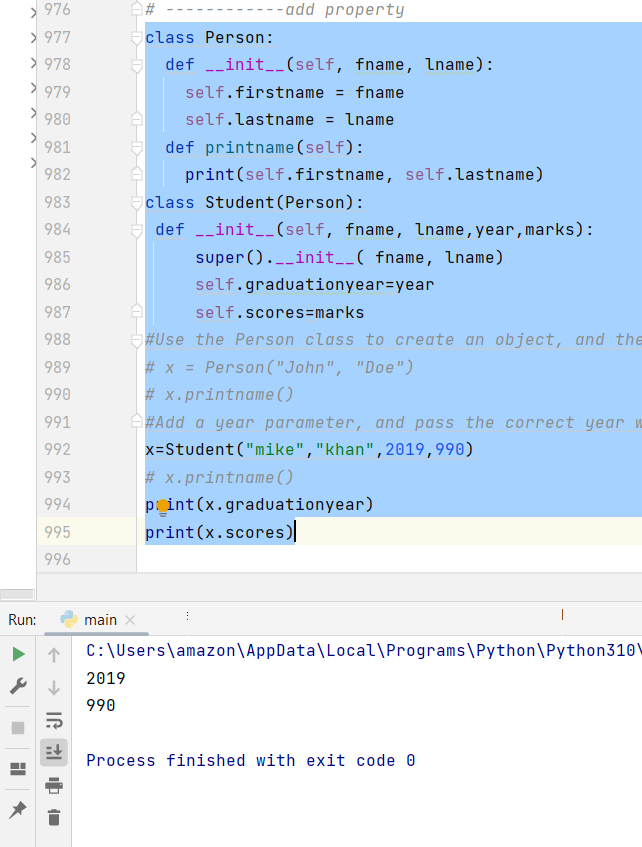
class Person:  
 def \_\_init\_\_(self, name, age):  
 self.name = name  
 self.age = age  
p1 = Person("John", 36)  
print(p1.name)  
print(p1.age)



class Person:  
 def \_\_init\_\_(self, fname, lname):  
 self.firstname = fname  
 self.lastname = lname  
 def printname(self):  
 print(self.firstname, self.lastname)  
class Student(Person):  
 def \_\_init\_\_(self, fname, lname):  
 super().\_\_init\_\_( fname, lname)  
*#Use the Person class to create an object, and then execute the printname method:  
# x = Person("John", "Doe")  
# x.printname()*x=Student("mike","khan")  
x.printname()



class Person:  
 def \_\_init\_\_(self, fname, lname):  
 self.firstname = fname  
 self.lastname = lname  
 def printname(self):  
 print(self.firstname, self.lastname)  
class Student(Person):  
 def \_\_init\_\_(self, fname, lname,year,marks):  
 super().\_\_init\_\_( fname, lname)  
 self.graduationyear=year  
 self.scores=marks  
*#Use the Person class to create an object, and then execute the printname method:  
# x = Person("John", "Doe")  
# x.printname()  
#Add a year parameter, and pass the correct year when creating objects:*x=Student("mike","khan",2019,990)  
*# x.printname()*print(x.graduationyear)  
print(x.scores)

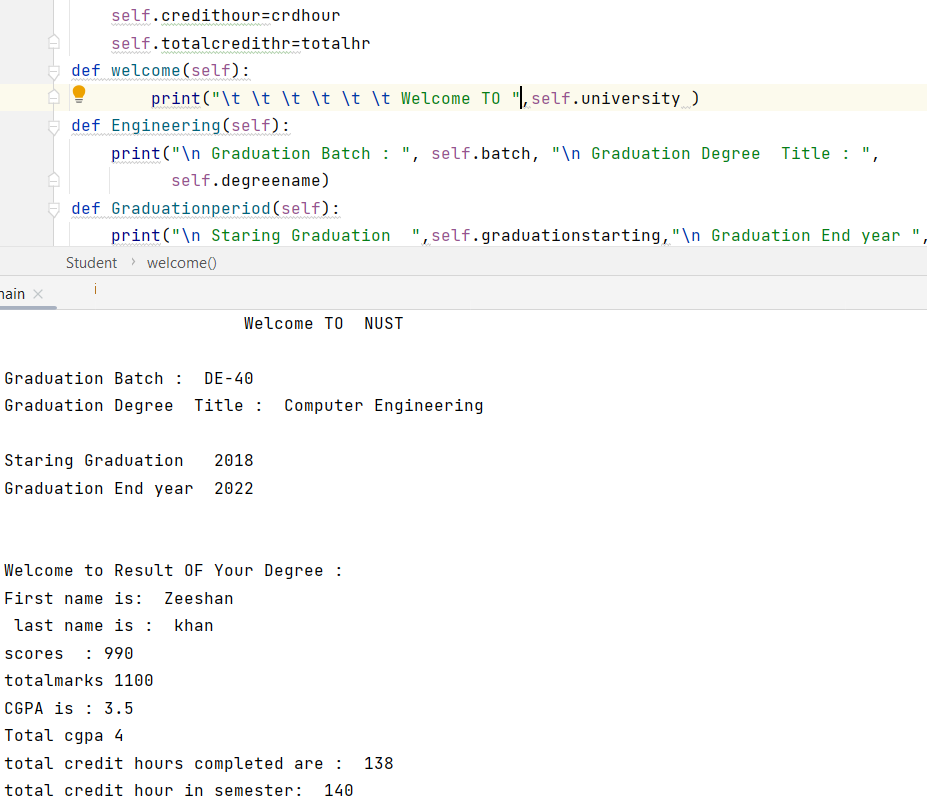
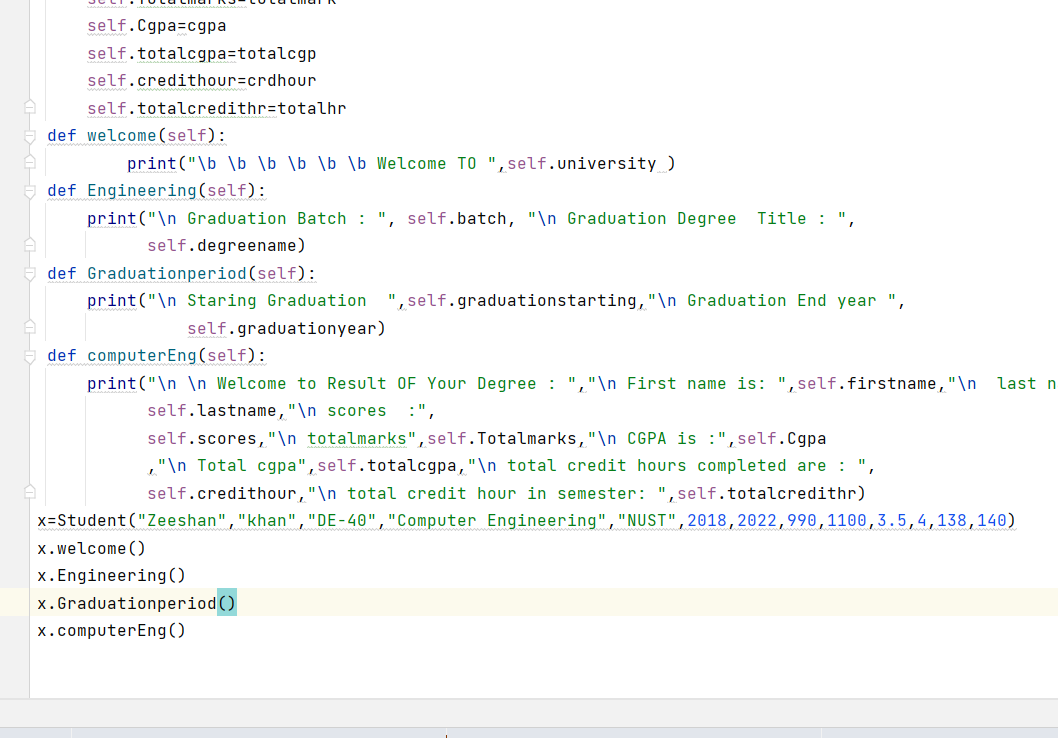
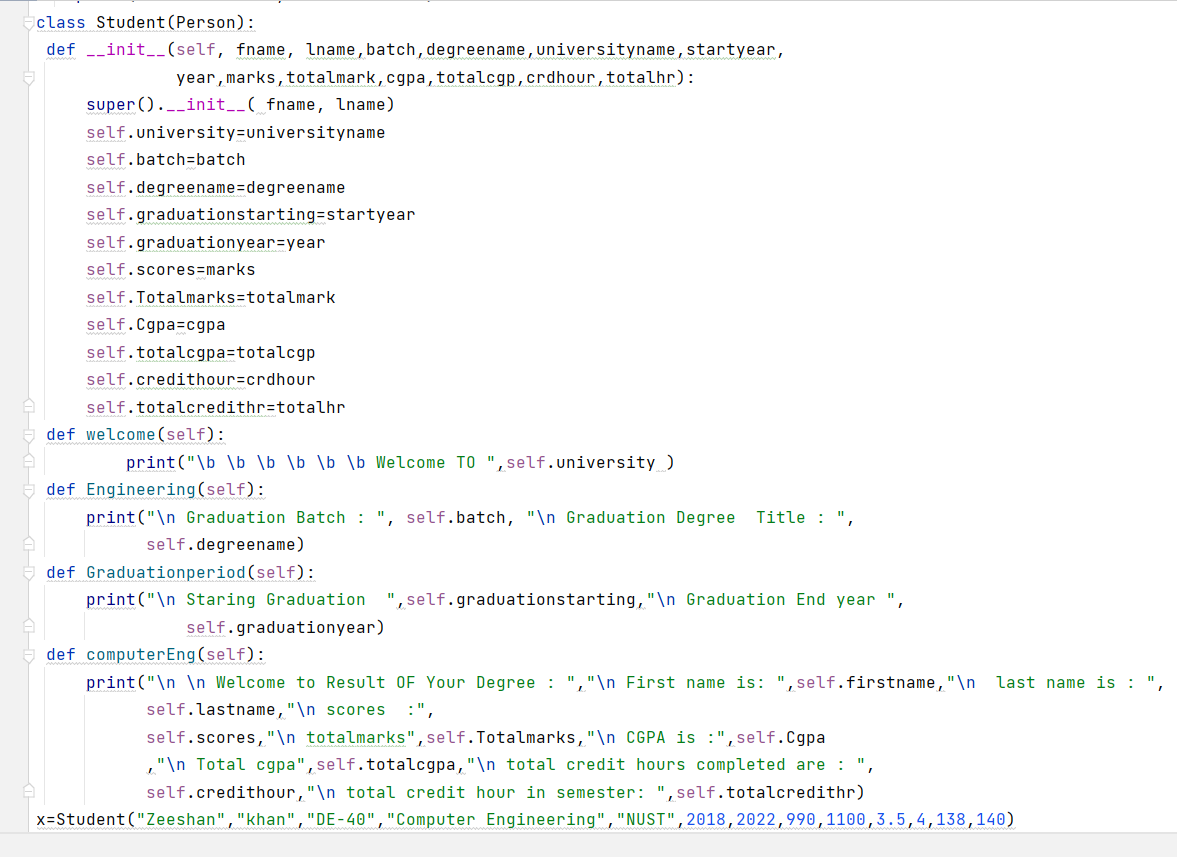
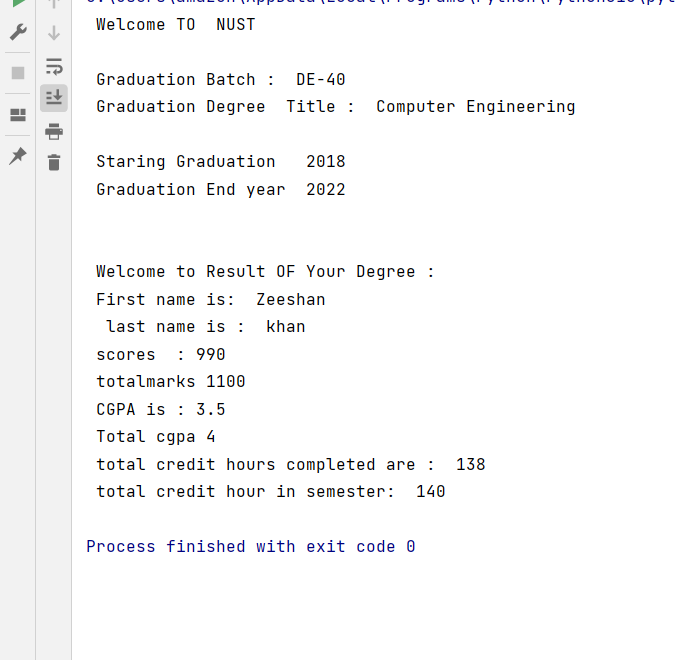


class Person:  
 def \_\_init\_\_(self, fname, lname):  
 self.firstname = fname  
 self.lastname = lname  
 def printname(self):  
 print(self.firstname, self.lastname)  
class Student(Person):  
 def \_\_init\_\_(self, fname, lname,year,marks):  
 super().\_\_init\_\_( fname, lname)  
 self.graduationyear=year  
 self.scores=marks  
 def welcome(self):  
 print("Welcome",self.firstname ," ",self.lastname,"To the class of ",self.graduationyear," with Marks",self.scores)  
*#Use the Person class to create an object, and then execute the printname method:  
# x = Person("John", "Doe")  
# x.printname()  
#Add a year parameter, and pass the correct year when creating objects:*x=Student("mike","khan",2019,990)  
*# x.printname()  
# print(x.graduationyear)  
# print(x.scores)*x.welcome()

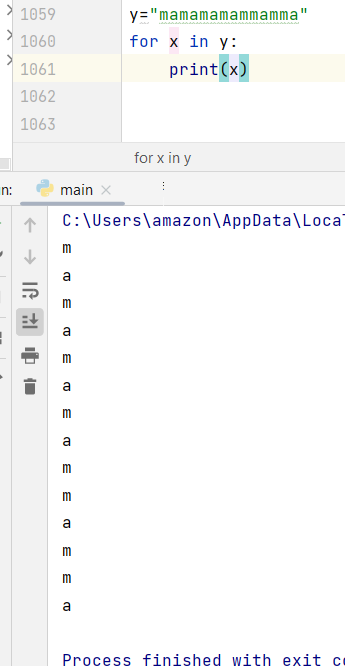
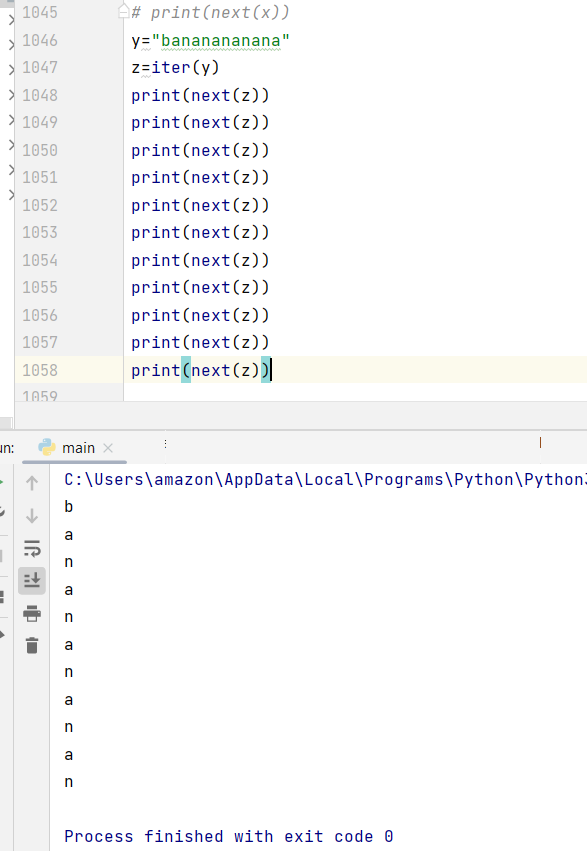
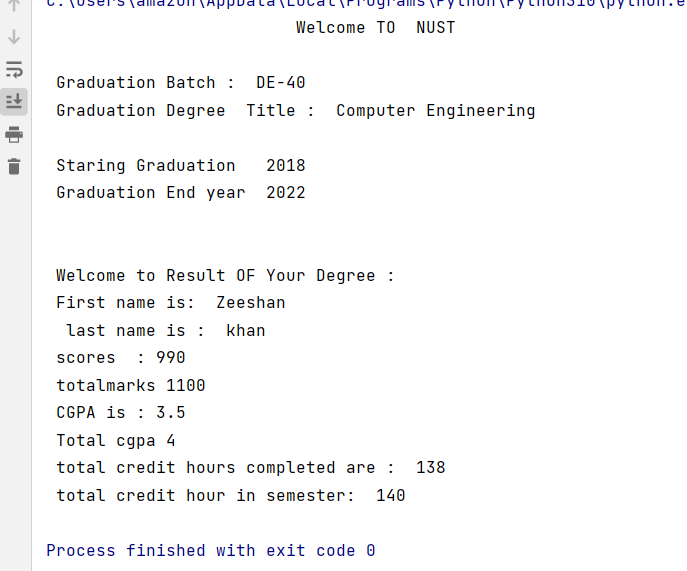


**Graduation Class Nust Inheritance :**

class Person:  
 def \_\_init\_\_(self, fname, lname):  
 self.firstname = fname  
 self.lastname = lname  
 def printname(self):  
 print(self.firstname, self.lastname)  
class Student(Person):  
 def \_\_init\_\_(self, fname, lname,batch,degreename,universityname,startyear,  
 year,marks,totalmark,cgpa,totalcgp,crdhour,totalhr):  
 super().\_\_init\_\_( fname, lname)  
 self.university=universityname  
 self.batch=batch  
 self.degreename=degreename  
 self.graduationstarting=startyear  
 self.graduationyear=year  
 self.scores=marks  
 self.Totalmarks=totalmark  
 self.Cgpa=cgpa  
 self.totalcgpa=totalcgp  
 self.credithour=crdhour  
 self.totalcredithr=totalhr  
 def welcome(self):  
 print("\b \b \b \b \b \b Welcome TO ",self.university )  
 def Engineering(self):  
 print("\n Graduation Batch : ", self.batch, "\n Graduation Degree Title : ",  
 self.degreename)  
 def Graduationperiod(self):  
 print("\n Staring Graduation ",self.graduationstarting,"\n Graduation End year ",  
 self.graduationyear)  
 def computerEng(self):  
 print("\n \n Welcome to Result OF Your Degree : ","\n First name is: ",self.firstname,"\n last name is : ",  
 self.lastname,"\n scores :",  
 self.scores,"\n totalmarks",self.Totalmarks,"\n CGPA is :",self.Cgpa  
 ,"\n Total cgpa",self.totalcgpa,"\n total credit hours completed are : ",  
 self.credithour,"\n total credit hour in semester: ",self.totalcredithr)  
x=Student("Zeeshan","khan","DE-40","Computer Engineering","NUST",2018,2022,990,1100,3.5,4,138,140)  
x.welcome()  
x.Engineering()  
x.Graduationperiod()  
x.computerEng()



class Person:  
 def \_\_init\_\_(self, fname, lname):  
 self.firstname = fname  
 self.lastname = lname  
 def printname(self):  
 print(self.firstname, self.lastname)  
class Student(Person):  
 def \_\_init\_\_(self, fname, lname,batch,degreename,universityname,startyear,  
 year,marks,totalmark,cgpa,totalcgp,crdhour,totalhr):  
 super().\_\_init\_\_( fname, lname)  
 self.university=universityname  
 self.batch=batch  
 self.degreename=degreename  
 self.graduationstarting=startyear  
 self.graduationyear=year  
 self.scores=marks  
 self.Totalmarks=totalmark  
 self.Cgpa=cgpa  
 self.totalcgpa=totalcgp  
 self.credithour=crdhour  
 self.totalcredithr=totalhr  
 def welcome(self):  
 print("\t \t \t \t \t \t Welcome TO ",self.university )  
 def Engineering(self):  
 print("\n Graduation Batch : ", self.batch, "\n Graduation Degree Title : ",  
 self.degreename)  
 def Graduationperiod(self):  
 print("\n Staring Graduation ",self.graduationstarting,"\n Graduation End year ",  
 self.graduationyear)  
 def computerEng(self):  
 print("\n \n Welcome to Result OF Your Degree : ","\n First name is: ",self.firstname,"\n last name is : ",  
 self.lastname,"\n scores :",  
 self.scores,"\n totalmarks",self.Totalmarks,"\n CGPA is :",self.Cgpa  
 ,"\n Total cgpa",self.totalcgpa,"\n total credit hours completed are : ",  
 self.credithour,"\n total credit hour in semester: ",self.totalcredithr)  
x=Student("Zeeshan","khan","DE-40","Computer Engineering","NUST",2018,2022,990,1100,3.5,4,138,140)  
x.welcome()  
x.Engineering()  
x.Graduationperiod()  
x.computerEng()



class Person:  
 def \_\_iter\_\_(self):  
 self.a=1  
 return self  
 def \_\_next\_\_(self):  
 if self.a<=20:  
 x=self.a  
 self.a+=1  
 return x  
 else:  
 raise StopIteration  
myclass=Person()  
myiter=iter(myclass)  
for x in myiter:  
 print(x)

