**CO4 Programs**

1. Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

**Program**

class rectangle:

def \_\_init\_\_(self,length,breadth):

self.length=length

self.breadth=breadth

def perimeter(self):

per=2\*(self.length+self.breadth)

return per

def area(self):

ar=(self.length\*self.breadth)

return ar

print("Enter the Length & Breadth of 1st rectangle")

l1=int(input("Length of rectangle: "))

b1=int(input("Breadth of rectangle: "))

p1=rectangle(l1,b1)

print("\nArea of 1st rectangle: ",p1.area())

print("Perimeter of 1st rectangle: ",p1.perimeter())

print("\nEnter the Length & Breadth of 2nd rectangle")

l2=int(input("Length of rectangle: "))

b2=int(input("Breadth of rectangle: "))

p2=rectangle(l2,b2)

print("\nArea of 2nd rectangle: ",p2.area())

print("Perimeter of 2nd rectangle: ",p2.perimeter())

if(p1.area()>p2.area()):

print("Area of Rectangle 1 is greater")

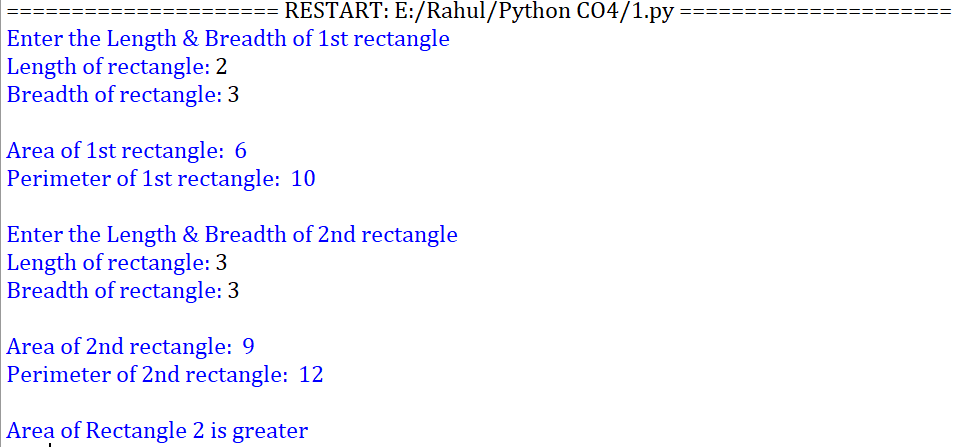
elif(p2.area()>p1.area()):

print("Area of Rectangle 2 is greater")

else:

print("\nBoth are equal")

**Output**



1. Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

**Program**

class bank:

def \_\_init\_\_(self,acno,name,atype,bal=0):

self.acno=acno

self.name=name

self.atype=atype

self.bal=bal

def deposit(self,bal):

self.bal=self.bal+bal

def withdraw(self,bal):

if(self.bal==0):

print("Insufficient balance!")

else:

self.bal=self.bal-bal

def display(self):

print("Name: ",self.name)

print("AC No : ",self.acno)

print("Account typle: ",self.atype)

print("Balance amount: ",self.bal)

acno=int(input("Enter the account number: "))

name=input("Enter the customer name: ")

atype=input("Enter the account type(Savings/Current): ")

bn=bank(acno,name,atype)

while(True):

print("ENTER YOUR CHOICE")

print()

print("1.Withdraw")

print("2.Deposit")

print("3.Display")

print("4.Exit")

c=int(input())

if(c==1):

a=int(input("Enter the amount to withdraw: "))

bn.withdraw(a)

if(c==2):

a2=int(input("Enter the amount to deposit: "))

bn.deposit(a2)

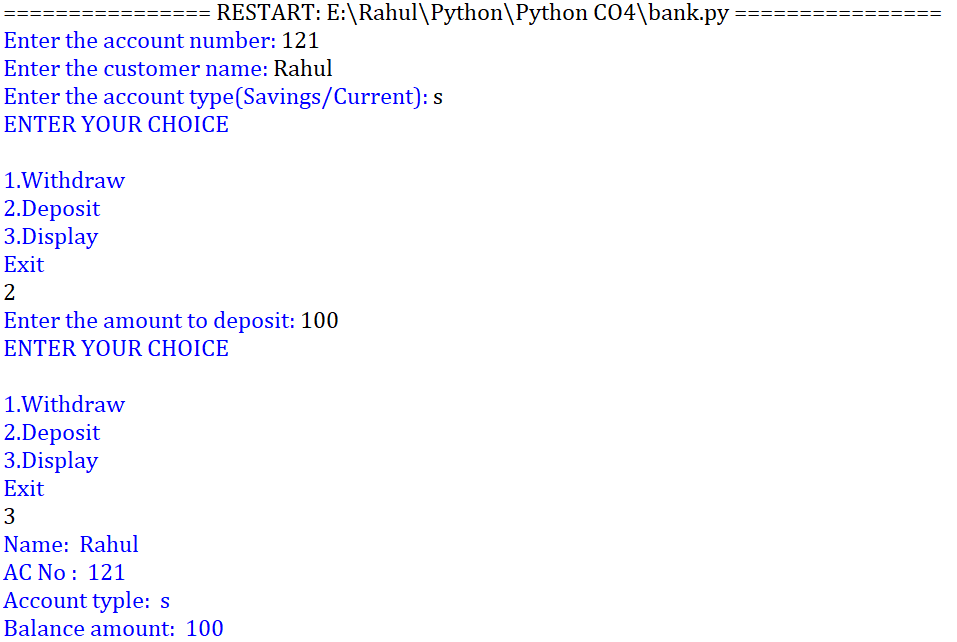
if(c==3):

bn.display()

if(c==4):

break

**Output**



1. Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.

**Program**

class rect:

def \_\_init\_\_(self,l,b):

self.\_\_l=l

self.\_\_b=b

def \_\_lt\_\_(self,a):

a1=self.\_\_l\*self.\_\_b

a2=a.\_\_l\*a.\_\_b

if(a1<a2):

return(True)

print("Rectangle")

l1=int(input("Enter the length: "))

b1=int(input("Enter the breadth: "))

print("Rectangle 2")

l2=int(input("Enter the length: "))

b2=int(input("Enter the breadth: "))

ar1=rect(l1,b1)

ar2=rect(l2,b2)

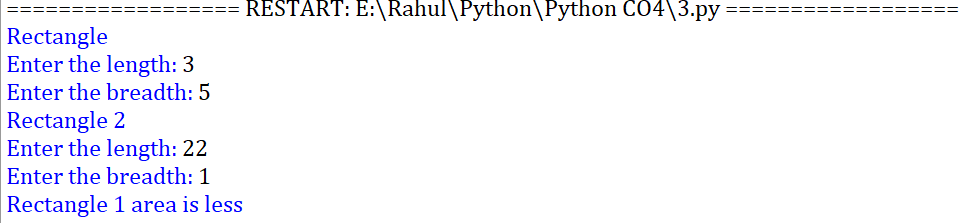
if(ar1<ar2):

print("Rectangle 1 area is less")

else:

print("Retangle 2 area is less")

**Output**



1. Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.

**Program**

class time:

def \_\_init\_\_(self,hr,mins,sec):

self.\_\_hr=hr

self.\_\_mins=mins

self.\_\_sec=sec

def \_\_add\_\_(self,a):

h=self.\_\_hr+a.\_\_hr

m=self.\_\_mins+a.\_\_mins

s=self.\_\_sec+a.\_\_sec

print(h,":",m,":",s)

print("Time 1")

h1=int(input("Enter hour: "))

m1=int(input("Enter minute: "))

s1=int(input("Enter second: "))

tm1=time(h1,m1,s1)

print("Time 2")

h2=int(input("Enter hour: "))

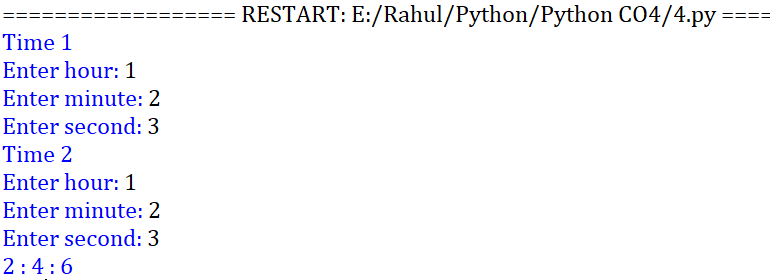
m2=int(input("Enter minute: "))

s2=int(input("Enter second: "))

tm2=time(h2,m2,s2)

tm1+tm2

**Output**



1. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

**Program**

class publisher:

def \_\_init\_\_(self,title,author):

self.title=title

self.author=author

def display(self):

print("Title:",self.title)

print("Author:",self.author)

class book(publisher):

def \_\_init\_\_(self,price,no\_of\_page):

self.price=price

self.no\_of\_page=no\_of\_page

def display(self):

print("Price:",self.price)

print("No. of Pages:",self.no\_of\_page)

class python(book):

def \_\_init\_\_(self,title,author,price,no\_of\_page):

publisher.\_\_init\_\_(self,title,author)

book.\_\_init\_\_(self,price,no\_of\_page)

def display(self):

print("Title:",self.title)

print("Author:",self.author)

print("Price:",self.price)

print("No. of Pages:",self.no\_of\_page)

p=python("Python Programming","Ashok Namdev Kamthane",500,100)

p.display()

**Output**

Title: Python Programming

Author: Ashok Namdev Kamthane

Price: 500

No. of Pages: 100