**CO4**

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

class Rectangle:

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

self.length=length

self.breadth=breadth

self.ar=0

def area(self):

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

return (self.ar)

def perimeter(self):

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

return (self.perimeter)

def display(self):

print("area=",self.ar)

print("perimeter=",self.perimeter)

R1=Rectangle(2,4,0)

R2=Rectangle(3,4,0)

R1.area()

R1.perimeter()

R2.area()

R2.perimeter()

print("Area of Rectangle1")

R1.display()

print("Area of Rectangle2")

R2.display()

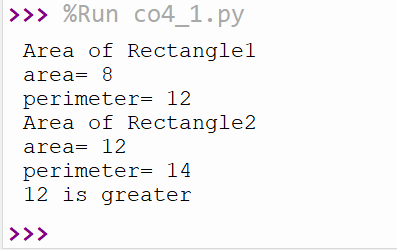
if (R1.ar>R2.ar):

print(R1.ar,"is graeter")

else:

print(R2.ar,"is greater")

OUTPUT



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

**2. 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.**

class Bank:

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

#self.accno=accno

#self.name=name

#self.acctype=acctype

self.bal=bal

name=input("Enter name:")

print("Account for",name,"is created")

def deposit(self):

amount=int(input("Amount to deposit"))

self.bal=self.bal+amount

print("New balance:",self.bal)

def withdarw(self):

amount=int(input("Amount to withdraw"))

if(self.bal>amount):

self.bal=self.bal-amount

print("New balance:",self.bal)

else:

print("insufficient amount")

print("balance:",self.bal)

def display(self):

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

print("account")

b1=Bank()

opt='y'

while(opt=='y'):

#print("your choice: 1. deposit \n 2. withdarw \n 3. display\n")

choice=int(input("your choice: a. deposit \n 2. withdarw \n 3. display\n"))

if(choice == 1):

b1.deposit()

elif(choice==2):

b1.withdarw()

elif(choice==3):

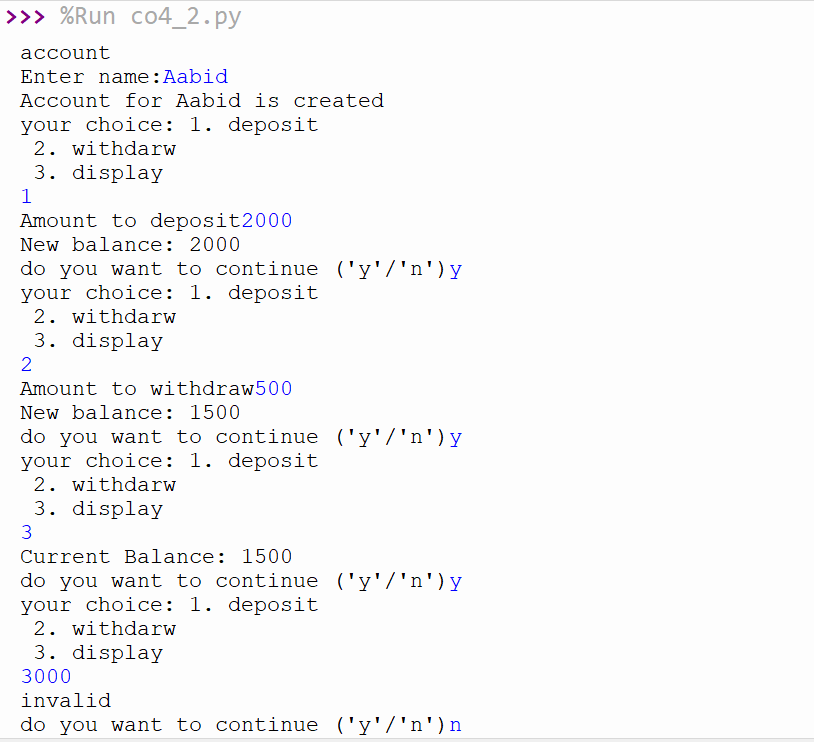
b1.display()

else:

print("invalid")

opt=input("do you want to continue ('y'/'n')")

OUTPUT



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

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

class rectangle:

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

self.length=length

self.width=width

def \_\_lt\_\_(self,a1):

area1=self.length\*self.width

area2=a1.length\*a1.width

if(area1>area2):

return(True)

else:

return(False)

print("Enter the Details of Rectangle:1")

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

w1=int(input("width:"))

r1=rectangle(l1,w1)

print("Enter the Details of Rectangle:2")

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

w2=int(input("width:"))

r2=rectangle(l2,w2)

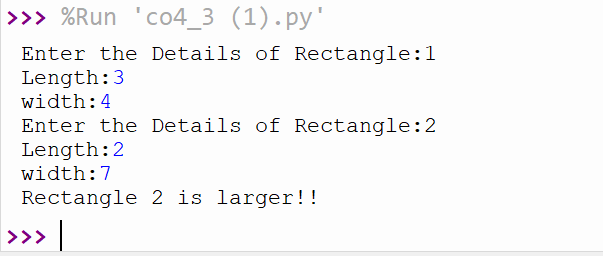
if(r1>r2):

print("Rectangle 2 is larger!!")

else:

print("Rectangle 1 is larger!!")

OUTPUT



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

**4. Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum**

**of 2 time.**

class Time:

def \_\_init\_\_(self,hour,minute,second):

self.\_\_hour=hour

self.\_\_minute=minute

self.\_\_second=second

def \_\_add\_\_(self,a2):

second=self.\_\_second+a2.\_\_second

minute=self.\_\_minute+a2.\_\_minute

hour=self.\_\_hour+a2.\_\_hour

if(second>60):

second=second-60

minute=minute+1

if(minute>60):

minute=minute-60

hour=hour+1

return hour,minute,second

print("Enter time1:")

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

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

s1=int(input("second"))

t1=Time(h1,m1,s1)

print("Enter time2:")

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

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

s2=int(input("second"))

t2=Time(h2,m2,s2)

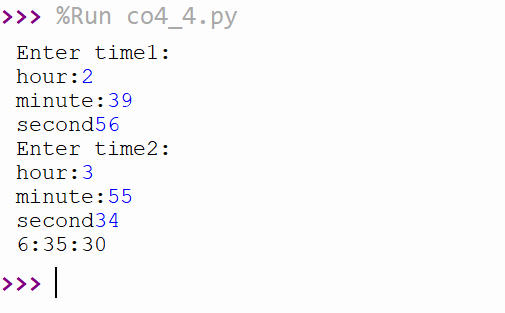
hr,min,sec=t1+t2

print(hr,end=":")

print(min,end=":")

print(sec,end=" ")

OUTPUT



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

**5. 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.**

class publisher:

def \_\_init\_\_ (self,pn):

self.publishername=pn

def display(self):

print("Publisher Name:",self.publishername)

class book(publisher):

def \_\_init\_\_ (self,pn,tt,aut):

super(). \_\_init\_\_(pn)

self.title=tt

self.author=aut

def display(self):

super().display()

print("Title Name: ",self.title)

print("Author Name:",self.author)

sclass python(book):

def \_\_init\_\_ (self,pn,tt,aut,pr,pg):

super(). \_\_init\_\_(pn,tt,aut)

self.price=pr

self.page=pg

def pythondisplay(self):

print("Price: ",self.price)

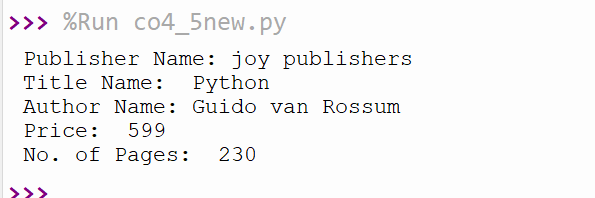
print("No. of Pages: ",self.page)

obj=python("joy publishers","Python","Guido van Rossum",599,230);

obj.display()

obj.pythondisplay();

OUTPUT



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