**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,l,b):

self.length=l

self.breadth=b

def area(self):

area=self.length\*self.breadth

print("Area of Rectangle: ",area)

return(area)

def perimeter(self):

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

print("Perimeter of Rectangle: ",perimeter)

print("Rectangle 1")

obj1=rectangle(40,20)

a1=obj1.area()

obj1.perimeter()

print("\nRectangle 2")

obj2=rectangle(30,20)

a2=obj2.area()

obj2.perimeter()

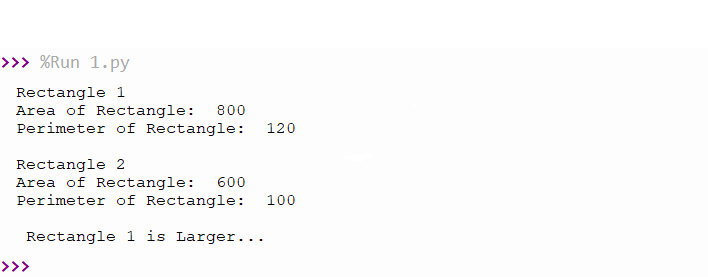
if a1 > a2:

print("\n Rectangle 1 is Larger...")

else:

print("\n Rectangle 2 is Larger...")

**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:

balance=0

def \_\_init\_\_(self,accountno,name,accounttype,balance):

self.accountno=accountno

self.name=name

self.accounttype=accounttype

self.balance=balance

def accountinformation(self):

print("\n --ACCOUNT INFORMATION--\n")

print("Account Number:",self.accountno)

print("Account Name:",self.name)

print("Account Type:",self.accounttype)

print("Account Balance:",self.balance,".00")

print("------------------------")

def deposit(self):

deposit=int(input("\n Enter the Amount to Deposit: "))

print("Rs.",deposit,"Deposited Successfully...")

print("------------------------")

self.balance=self.balance+deposit

def withdraw(self):

withdraw=int(input("\n Enter the Amount to Withdraw: "))

if withdraw > self.balance:

print("Your Account has Insufficient Balance...")

print("------------------------")

else:

self.balance=self.balance-withdraw

print("Rs.",withdraw,"Withdrawn Successfully...")

print("------------------------")

print(" Enter the Details of your Bank Account")

acc\_no=int(input("Enter the Account Number:"))

acc\_name=input("Enter the Name:")

acc\_type=input("Enter the Account type-(Savings/Current):")

balance=int(input("Enter the Initial Balance:"))

obj=bank(acc\_no,acc\_name,acc\_type,balance)

while(1):

print("\n --WELCOME TO PYTHON BANK--")

print("\n1.Account Information\n2.Deposit\n3.Withdraw\n4.Exit\n")

opt=int(input("Select your option:"))

if opt == 1:

obj.accountinformation()

elif opt == 2:

obj.deposit()

elif opt == 3:

obj.withdraw()

elif opt == 4:

print("Exited")

print(" Thank You Visit Again....\n")

print("------------------------")

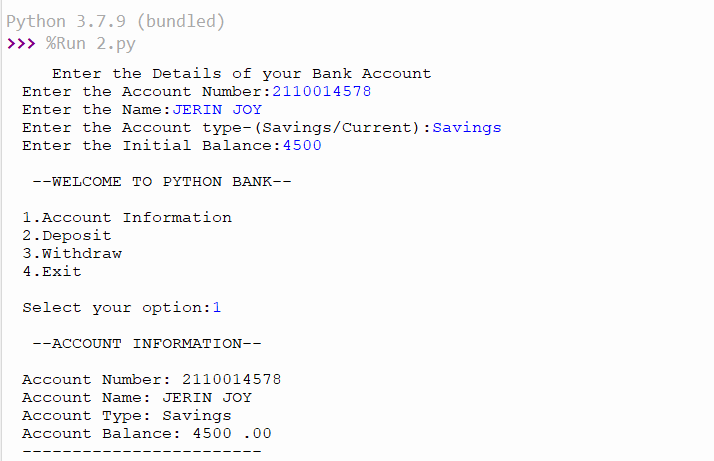
break

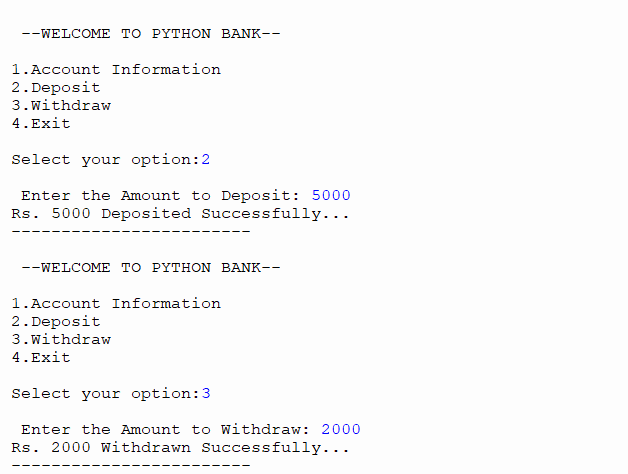
else:

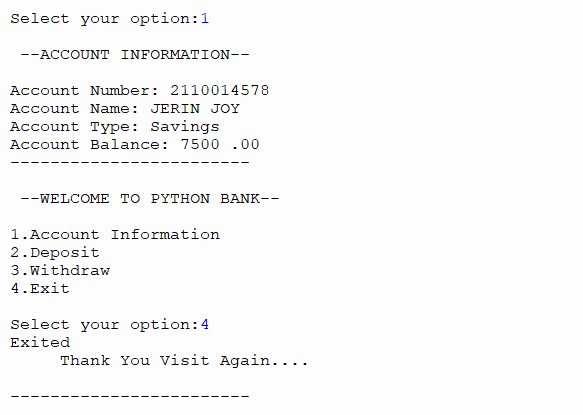
print("Invalid Option")

print("------------------------")

**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,l,b):

self.\_\_length=l

self.\_\_breadth=b

def area(self):

self.area=self.\_\_length\*self.\_\_breadth

print("Area of Rectangle: ",self.area)

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

if self.area < second.area:

return True

else:

return False

print("Rectangle 1")

length1=int(input("Enter the length:"))

breadth1=int(input("Enter the breadth:"))

obj1=rectangle(length1,breadth1)

obj1.area()

print("\nRectangle 2")

length2=int(input("Enter the length:"))

breadth2=int(input("Enter the breadth:"))

obj2=rectangle(length2,breadth2)

obj2.area()

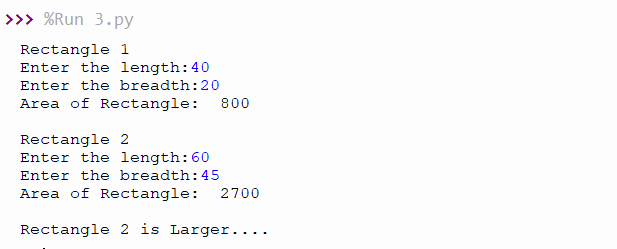
if obj1 > obj2 :

print("\nRectangle 1 is Larger.....")

else:

print("\nRectangle 2 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,hr,min,sec):

self.\_\_hour=hr

self.\_\_minute=min

self.\_\_second=sec

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

print("\nHOUR: ",self.\_\_hour + second.\_\_hour,"hour")

if self.\_\_minute + second.\_\_minute > 60:

hr1=(self.\_\_minute + second.\_\_minute)//60

min1=(self.\_\_minute + second.\_\_minute)%60

print("MINTUES:",h1," hour ",m1," minutes")

else:

print("MINTUES:",self.\_\_minute + second.\_\_minute,"mintues")

if self.\_\_second+second.\_\_second > 60:

min1=(self.\_\_second+second.\_\_second)//60

sec1=(self.\_\_second+second.\_\_second)%60

print("SECONDS:",m1," minutes ",s1," seconds")

else:

print("SECONDS:",self.\_\_second + second.\_\_second,"seconds")

hour1=int(input("Enter the hour:"))

minute1=int(input("Enter the minutes:"))

second1=int(input("Enter the second:"))

obj1=time(hour1,minute1,second1)

hour2=int(input("\nEnter the hour:"))

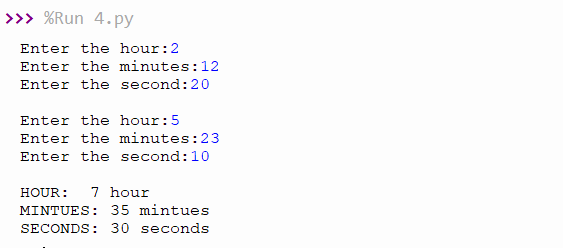
minute2=int(input("Enter the minutes:"))

second2=int(input("Enter the second:"))

obj2=time(hour2,minute2,second2)

obj1 + obj2

**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):

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

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

class 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)

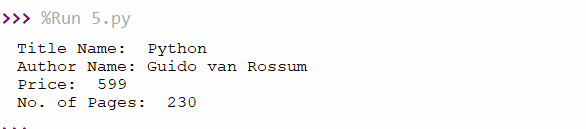
print("No. of Pages: ",self.page)

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

obj.display()

obj.pythondisplay();

**OUTPUT**



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