**CO4 PROGRAM**

**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,breadth,length):

self.breadth=breadth

self.length=length

def area(self):

return self.breadth\*self.length

def perimeter(self):

return 2\*self.breadth+self.length

a1=int(input("Enter length of rectangle: "))

b1=int(input("Enter breadth of rectangle: "))

obj1=rectangle(a1,b1)

print("Area of rectangle:",obj1.area())

print("perimeter of rectangle:",obj1.perimeter())

a2=int(input("Enter length of rectangle: "))

b2=int(input("Enter breadth of rectangle: "))

obj2=rectangle(a2,b2)

print("Area of rectangle:",obj2.area())

print("perimeter of rectangle:",obj2.perimeter())

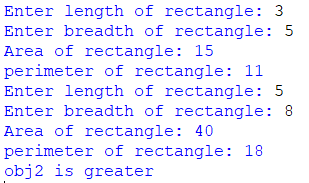
if (obj1.area()>obj2.area()):

print("obj1 is greater")

else:

print("obj2 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):

self.balance=0

name=input("enter the name of account holder:")

acno=int(input("enter the account no:"))

print ("The account is created")

print("\n name of account:",name)

print("\n account no:",acno)

def deposit(self):

amount=int(input(" enter the amount:"))

self.balance+=amount

def withdraw(self):

amount = float(input("Enter amount to be Withdrawn:"))

if (self.balance>=amount):

self.balance-=amount

print("\nYou Withdraw:", amount)

else:

print("\ninsufficient balance")

def display(self):

print("\nAvailable Balance =",self.balance)

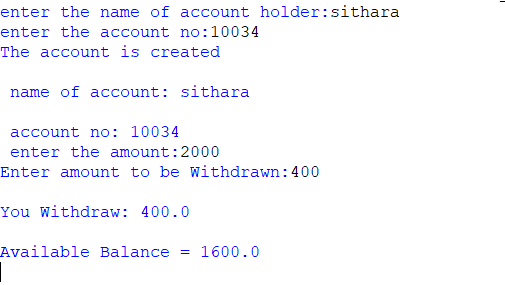
b=bank()

b.deposit()

b.withdraw()

b.display()

**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)

a1=int(input("Length of 1 rectangle:"))

b1=int(input("width 1 rectangle:"))

r1=rectangle(a1,b1)

a2=int(input("Length 2 rectangle:"))

b2=int(input("width 2 rectangle:"))

r2=rectangle(a2,b2)

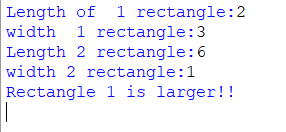
if(r1<r2):

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,h):

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

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

hour=self.\_\_hour+h.\_\_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 1 time:")

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

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

s1=int(input(" enter the second:"))

t1=Time(h1,m1,s1)

print("Enter 2 time:")

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

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

s2=int(input("enter the second:"))

t2=Time(h2,m2,s2)

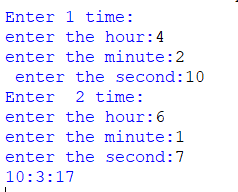
hr,min,sec=t1+t2

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,pname):

self.pname=pname

def display(self):

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

class book(publisher):

def get(self,title,author):

self.title=title

self.author=author

def display(self):

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

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

class python(book):

def \_\_init\_\_(self,price,nop,pname):

super().\_\_init\_\_(pname)

self.price=price

self.nop=nop

def details(self):

print("Price:",self.price)

print("No of pages:",self.nop)

s1=python(1200,192,"KIRAN")

s1.get("PYTHON PROGRAMMING","KIRAN")

s1.display()

s1.details()

**OUTPUT**

