**CO4 program**

**Q1 Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.**

classRectangle:

    def\_\_init\_\_(self):

        self.l=int(input("enter the length"))

        self.b=int(input("enter the breadth"))

        self.area=self.l\*self.b

        self.perimeter=2\*(self.l+self.b)

    defdisplay(self):

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

        print("Perimeter of Rectangle",self.perimeter)

print("Rectangle 1")

p1=Rectangle()

p1.display()

print("Rectangle 2")

p2=Rectangle()

p2.display()

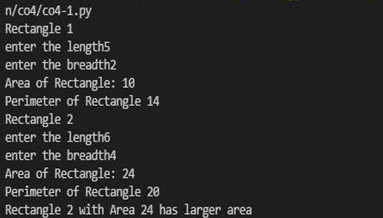
if p1.area>p2.area:

    print("Rectangle 1 with Area", p1.area, "has larger area")

else:

    print("Rectangle 2 with Area",p2.area,"has larger area")

**OUTPUT**

****

**Q2 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.**

classBank:

    def\_\_init\_\_(self):

        self.bal=0

    defcreate(self) :

        print("enter The Details")

        self.no=int(input("enter the account Number"))

        self.name=input("enter the name")

        self.type=input("Entr the account type")

    defdeposit(self):

        self.amount=int(input("enter the amount to deposit"))

        self.bal=self.bal+self.amount

        print("BALANCE",self.bal)

    defwithdraw(self):

        self.amount=int(input("enter the amount to withdraw"))

        self.bal=self.bal-self.amount

        print("BALANCE",self.bal)

    defdisplay(self):

        print("ACCOUNT NUMBER:",self.no)

        print("ACCOUNT HOLDER NAME:",self.name)

        print("ACCOUNT TYPE:",self.type)

        print("ACCOUNT BALANCE:",self.bal)

B=Bank()

B.create()

x=1

whilex!=0:

    print("MENU")

    x=int((input("1.DEPOSIT 2.WITHDRAW 3.BALANCE")))

    if x==1:

        B.deposit()

    elif x==2:

        B.withdraw()

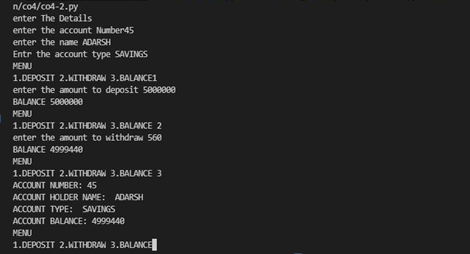
    elif x==3:

        B.display()

    else:

        print("invalid selection")

**OUTPUT**



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

classrectangle:

    def\_\_init\_\_(self):

        self.\_\_length=int(input("Enter the length: "))

        self.\_\_width=int(input("Enter the breadth: "))

    def\_\_lt\_\_(self,a1):

        area1=self.\_\_length\*self.\_\_width

        area2=a1.\_\_length\*a1.\_\_width

        if(area1<area2):

            return(True)

        else:

            return(False)

print("RECTANGLE 1")

r1=rectangle()

print("RECTANGLE 2")

r2=rectangle()

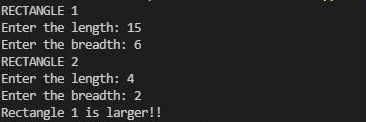
if(r1<r2):

    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.**

classTime:

    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

        if(hour>24):

            hour=hour-24

        returnhour,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.**

classpublisher:

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

        self.title=title

        self.author=author

    defdisplay(self):

        print("Title:",self.title)

        print("Author:",self.author)

classbook(publisher):

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

        self.price=price

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

    defdisplay(self):

        print("Price:",self.price)

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

classpython(book):

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

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

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

    defdisplay(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","m mukundhan",2000,200)

p.display()

**OUTPUT**

****