**PYTHON**

**CO-4 PROGRAMS**

**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

a=int(input("Enter length of First rectangle: "))

b=int(input("Enter breadth of First rectangle: "))

obj1=rectangle(a,b)

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

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

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

c=int(input("Enter length of Second rectangle: "))

d=int(input("Enter breadth of Second rectangle: "))

obj2=rectangle(c,d)

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

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

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

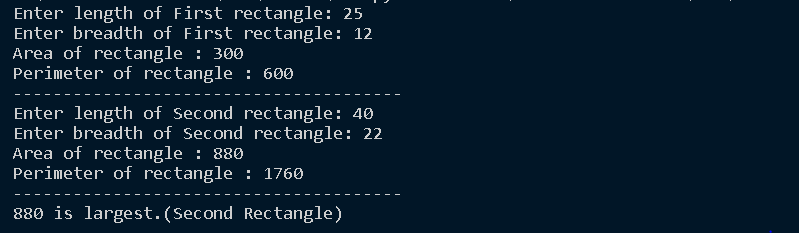
*if*(obj1.area()>obj2.area()):

    print(obj1.area(), "is largest.(First Rectangle)")

*else*:

    print(obj2.area(), "is largest.(Second Rectangle)")

**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 ("\n---The account is created---")

        print("\nName of Account Holder : ",name)

        print("\nAccount no : ",acno)

    def *deposit*(self):

        amount=int(input("\nEnter the amount to deposit : "))

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        amount = float(input("Enter amount to be Withdrawn : "))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

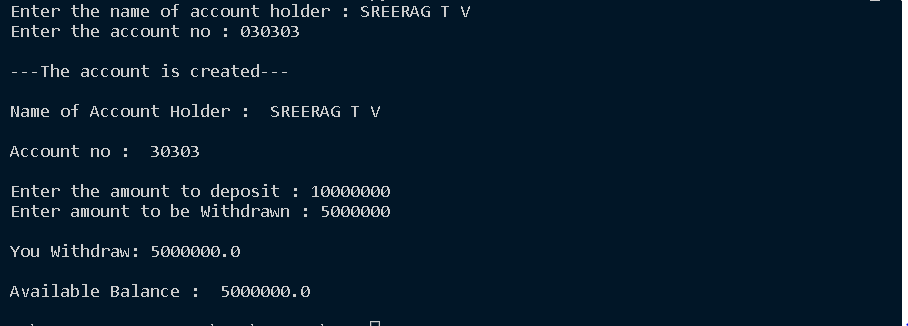
b=bank()

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  First Rectangle:"))

b1=int(input("Width of First Rectangle:"))

r1=rectangle(a1,b1)

a2=int(input("Length of Second Rectangle:"))

b2=int(input("Width of Second Rectangle:"))

r2=rectangle(a2,b2)

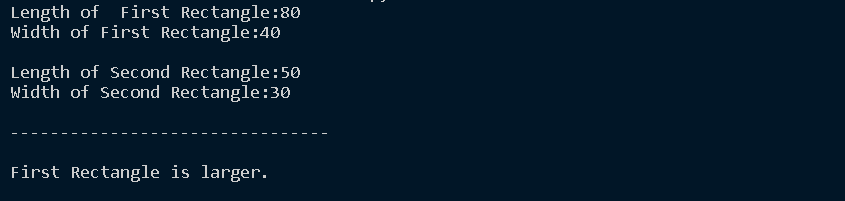
*if*(r1<r2):

    print("Second Rectangle is larger.")

*else*:

    print("First Rectangle 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 First Time---\n")

h1=int(input("Enter The Hour : "))

m1=int(input("Enter The Minute : "))

s1=int(input("Enter The Second : "))

t1=Time(h1,m1,s1)

print("\n---Enter Second Time---\n")

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

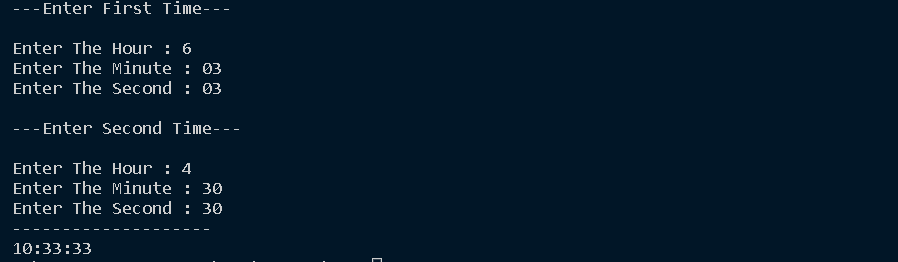
print("--------------------")

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(200,180,"A P J Abdul kalam")

s1.get("Wings Of Fire","A P J Abdul kalam")

s1.display()

s1.details()

**OUTPUT**

