**CO4 PROGRAMS**

**1. Create Rectangle class with attributes length and breadth and methods to find area and**

**perimeter. Compare two Rectangle objects by their area.**

***Input:***

class Rectangle:

def \_\_init\_\_(self,l,b):

self.l=l

self.b=b

self.area=self.l\*self.b

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

def display(self):

print("AREA OF RECTANGLE:",self.area)

print("PERIMETER OF RECTANGLE:",self.p)

p1=Rectangle(2,2)

p2=Rectangle(6,4)

print("RECTANGLE1")

p1.display()

print("RECTANGLE2")

p2.display()

if p1.area>p2.area:

print("LARGEST AREA",p1.area,)

else:

print("LARGEST",p2.area)

RECTANGLE1

AREA OF RECTANGLE: 4

PERIMETER OF RECTANGLE: 8

RECTANGLE2

AREA OF RECTANGLE: 24

PERIMETER OF RECTANGLE: 20

LARGEST 24

**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.**

***Input:***

class bank:

\_\_acc\_name=""

\_\_acc\_no = ""

\_\_acc\_type = ""

\_\_acc\_balance = 0

def \_\_init\_\_(self,a\_name,a\_no,a\_type,a\_balance):

self.\_\_acc\_name = a\_name

self.\_\_acc\_no = a\_no

self.\_\_acc\_type = a\_type

self.\_\_acc\_balance = a\_balance

def deposit(self,a\_deposit):

print("Initial balance is : ",self.\_\_acc\_balance)

print("Deposit is : ",a\_deposit)

self.\_\_acc\_balance += a\_deposit

print("Current balance: ",self.\_\_acc\_balance)

def withdraw(self):

print("Current balance: ",self.\_\_acc\_balance)

self.amount = int(input("Amount to withdraw :"))

if self.amount > self.\_\_acc\_balance:

print("You don't have enough balance to withdraw !!")

print("Current balance:",self.\_\_acc\_balance)

else:

print(self.amount," is withrawed .")

self.\_\_acc\_balance -= self.amount

print("Current balance:",self.\_\_acc\_balance)

def acc\_info(self):

print("Account holder name:",self.\_\_acc\_name)

print("Account number:",self.\_\_acc\_no)

print("Account type :",self.\_\_acc\_type)

print("Account Balance:",self.\_\_acc\_balance)

def main():

name = input("Account holder name : ")

no = input("Account number:")

atype = input("Account type:")

bal = int(input("Account initial balance:"))

holder = bank(name,no,atype,bal)

while(True):

opt = int(input("1)Deposit \n2)Withdraw \n3)Account info \n0)Exit\nChoose your option :: "))

if opt == 1:

amount = int(input("Deposit amount:"))

holder.deposit(amount)

elif opt == 2:

holder.withdraw()

elif opt == 3:

holder.acc\_info()

elif opt == 0:

break

else:

print("Invalid Option !")

if \_\_name\_\_ == "\_\_main\_\_":

while(True):

main()

***Output:***

Account holder name : gopika

Account number:78969

Account type:fixed

Account initial balance:10000

1)Deposit

2)Withdraw

3)Account info

0)Exit

Choose your option :: 1

Deposit amount:2000

Initial balance is : 10000

Deposit is : 2000

Current balance: 12000

1)Deposit

2)Withdraw

3)Account info

0)Exit

Choose your option :: 3

Account holder name: gopika

Account number: 78969

Account type : fixed

Account Balance: 12000

1)Deposit

2)Withdraw

3)Account info

0)Exit

Choose your option :: 2

Current balance: 12000

Amount to withdraw :1000

1000 is withrawed .

Current balance: 11000

1)Deposit

2)Withdraw

3)Account info

0)Exit

Choose your option :: 3

Account holder name: gopika

Account number: 78969

Account type : fixed

Account Balance: 11000

1)Deposit

2)Withdraw

3)Account info

0)Exit

Choose your option ::

**(3). Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to**

**compare the area of 2 rectangles.**

***Input:***

class rectangle:

\_\_area = 0

\_\_perimeter = 0

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

self.\_\_length = length

self.\_\_width = width

def calc\_area(self):

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

print("Area is :",self.\_\_area)

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

if self.\_\_area < second.\_\_area:

return True

else:

return False

length1= int(input("Enter length of the rectangle 1 : "))

width1 = int(input("Enter width of the rectangle 1 : "))

length2 = int(input("Enter length of the rectangle 2 : "))

width2 = int(input("Enter width of the rectangle 2 : "))

obj1 = rectangle(length1,width1)

obj2 = rectangle(length2,width2)

obj1.calc\_area()

obj2.calc\_area()

if obj1 < obj2:

print("Rectangle two is large")

else:

print("Rectangle one is large or these are equal")

***Output:***

Enter length of the rectangle 1 : 2

Enter width of the rectangle 1 : 3

Enter length of the rectangle 2 : 4

Enter width of the rectangle 2 : 2

Area is : 6

Area is : 8

Rectangle two is large

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

**find sum of 2 time.**

***Input:***

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:

Enter time1:

hour:2

minute:3

second4

Enter time2:

hour:3

minute:5

second7

5:8:11

**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.**

***Input:***

class publisher:

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

self.title=title

self.author=author

def display(self):

print("Title:",self.title)

print("Author:",self.author)

class book(publisher):

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

self.price=price

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

def display(self):

print("Price:",self.price)

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

class python(book):

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

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

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

def display(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","ezhuthachan",1000,1000)

p.display()

***Output:***

Title: Python Programming

Author: ezhuthachan

Price: 1000

No. of Pages: 1000