Co4

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

Program

class rectangle:

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

self.l=l

self.b=b

def area(self):

area=self.l\*self.b

print("area of rectangle",area)

return(area)

def perimeter(self):

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

print("perimeter of rectangle",per)

return(per)

r1=rectangle(7,8)

r2=rectangle(8,7)

a=r1.area()

r1.perimeter()

b=r2.area()

r2.perimeter()

if(a>b):

print("Rectangle one area is greater",a)

else:

print("Rectangle two area is greater",b)

output

area of rectangle 56

perimeter of rectangle 30

area of rectangle 56

perimeter of rectangle 30

Rectangle two area is greater 56

1. 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

Program

class bank:

bal=0

def \_\_init\_\_(self,accno,name,ac\_type,bal):

self.accno=accno

self.name=name

self.ac\_type=ac\_type

self.bal=bal

def display(self):

print("\nAccount info:")

print("Account number:",self.accno)

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

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

print("Account balance:",self.bal)

def deposit(self):

dep=int(input("Enter amount deposit:"))

self.bal=self.bal+dep

def withdraw(self):

w=int(input("Enter amount withdraw:"))

if w > self.bal:

print("Insufficient Balance")

else:

self.bal=self.bal-w

print("Rs",w,"Successfully Withdrawn")

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

acc\_name=input("Enter name:")

acc\_type=input("Enter account type(savings/current):")

balance=int(input("Enter initial balance:"))

b1=bank(acc\_no,acc\_name,acc\_type,balance)

while(1):

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

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

if opt == 1:

b1.display()

elif opt == 2:

b1.deposit()

elif opt == 3:

b1.withdraw()

elif opt == 4:

print("Exit")

break

else:

print("Invalid Option")

output

Enter Account Number:4567

Enter name:abhi

Enter account type(savings/current):savings

Enter initial balance:400

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account info:

Account number: 4567

Account name: abhi

Account type: savings

Account balance: 400

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:2

Enter amount deposit:5000

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account info:

Account number: 4567

Account name: abhi

Account type: savings

Account balance: 5400

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:3

Enter amount withdraw:3000

Rs 3000 Successfully Withdrawn

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account info:

Account number: 4567

Account name: abhi

Account type: savings

Account balance: 2400

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:4

Exit

1. Create a class Rectangle with private attributes length and width. Overload

Program

class rectangle:

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

self.\_\_length=l

self.\_\_breadth=b

def area(self):

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

print("Area=",self.area)

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

if self.area < second.area:

return True

else:

return False

print("first Rectangle:")

len1=int(input("Enter the length:"))

bread1=int(input("Enter the breadth:"))

obj1=rectangle(len1,bread1)

obj1.area()

print("\nSecond Rectangle:")

len2=int(input("Enter the length:"))

bread2=int(input("Enter the breadth:"))

obj2=rectangle(len2,bread2)

obj2.area()

if obj1 < obj2 :

print("\nArea of second rectangle is larger:")

else:

print("\nArea of first rectangle is larger:")

output

first Rectangle:

Enter the length:2

Enter the breadth:3

Area= 6

Second Rectangle:

Enter the length:5

Enter the breadth:6

Area= 30

Area of second rectangle is larger:

1. Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.

Program

class time:

def \_\_init\_\_(self,hour,minute,second):

self.\_\_hour=hour

self.\_\_minute=minute

self.\_\_second=second

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

print("Hour",self.\_\_hour+tm.\_\_hour)

print("Minute",self.\_\_minute+tm.\_\_minute)

print("Second",self.\_\_second+tm.\_\_second)

hr=int(input("Enter the hour"))

mins=int(input("Enter the min"))

sec=int(input("Enter the seco"))

obj=time(hr,mins,sec)

hr1=int(input("Enter the hour"))

mins1=int(input("Enter the min"))

sec1=int(input("Enter the seco"))

obj1=time(hr1,mins1,sec1)

obj+obj1

output

Enter the hour3

Enter the min45

Enter the seco23

Enter the hour6

Enter the min45

Enter the seco54

Hour 9

Minute 90

Second 77

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.

Program

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("java script","Brendan Eich",1000,120)

p.display()

output

Title: java script

Author: Brendan Eich

Price: 1000

No. of Pages: 120