COURSE OUTCOME 4

DATE : 27-11-2023

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

PROGRAM

```python
class Rectangle:
        def __init__(self,l=0,b=0):
                self.l=l
                self.b=b
        def area(self):
                return self.l*self.b
        def perimeter(self):
                return 2*(self.l+self.b)
        def compare(self, s):
                if(self.area()>s.area()):
                        print("Area of rectangle 1 is greater")
                else:
                        print("Area of rectangle 2 is greater")


print("Enter parameters of rectangle 1 : ")
l1=int(input("Enter length : "))
b1=int(input("Enter breadth : "))

print("Enter parameters of rectangle 2 : ")
l2=int(input("Enter length : "))
```

```python
b2=int(input("Enter breadth : "))

s1= Rectangle(l1,b1)
s2= Rectangle(l2,b2)

print("Area of rectangle 1 : ",s1.area())
print("Perimeter of rectangle 1 : ",s1.perimeter())


print("Area of rectangle 2 : ",s2.area())
print("Perimeter of rectangle 2 : ",s2.perimeter())

s1.compare(s2)
```

OUTPUT

Enter parameters of rectangle 1 :
Enter length : 5
Enter breadth : 10

Enter parameters of rectangle 2 :
Enter length : 15
Enter breadth : 20

Area of rectangle 1 : 50
Perimeter of rectangle 1 : 30

Area of rectangle 2 : 300
Perimeter of rectangle 2 : 70

Area of rectangle 2 is greater

DATE : 27-11-2023

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.

PROGRAM

```python
class Account:
    def __init__(self,acc_no,acc_name,acc_type,acc_balence):
        self.acc_no=acc_no
        self.acc_name=acc_name
        self.acc_type=acc_type
        self.acc_balence=acc_balence

    def deposit(self,amt):
        if amt>0:
            self.acc_balence=self.acc_balence+amt
            print("Rs ",amt,"Successfully Deposited")
        else:
            print("Inavlid Amount")

    def withdraw(self,amt):
        if amt>self.acc_balence:
            print("Insufficent Balence")
        else:
            print("Rs ",amt,"Successfully Withdrawn")
            self.acc_balence=self.acc_balence-amt
```

```python
    def view_details(self):
        print("Account No : ",self.acc_no,"\nName : ",self.acc_name,"\nType : ",self.acc_type,"\nBalence : ",self.acc_balence)

acc_no=int(input("Enter the account number : "))
acc_name=input("Enter the account holders name : ")
acc_type=input("Enter the account type : ")
acc_balence=int(input("Enter the account balance : "))

c1= Account(acc_no,acc_name,acc_type,acc_balence)

while True:
    print("1. Deposit\n2. Withdraw\n3. Current Balance\n4. View Details\n5. Exit")
    ch=int(input("Enter your choice : "))

    if ch==1:
        amt=int(input("Enter the amount to be deposited : "))
        c1.deposit(amt)
    elif ch==2:
        amt=int(input("Enter the amount to be withdrawn : "))
        c1.withdraw(amt)
    elif ch==3:
        print("Net Balance : ",c1.acc_balence)
    elif ch==4:
        c1.view_details()
    elif ch==5:
        break
```

OUTPUT

Enter the account number : 1001
Enter the account holders name : Anirudh J Bhatt
Enter the account type : Savings
Enter the account balance : 50000

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 1
Enter the amount to be deposited : 50000
Rs 50000 Successfully Deposited

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 3
Net Balance : 100000

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 2
Enter the amount to be withdrawn : 25000
Rs 25000 Successfully Withdrawn

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 3
Net Balance : 75000

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 4
Account No : 1001
Name : Anirudh J Bhatt
Type : Savings
Balance : 75000

1. Deposit
2. Withdraw
3. Current Balance
4. View Details
5. Exit
Enter your choice : 5

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

   PROGRAM

```
class Rectangle:
        def __init__(self, l, b):
                self.l=l
                self.b=b

        def area(self):
                return self.l*self.b

        def __lt__(self, s):
                return self.area()<s.area()

print("Enter parameters of rectangle 1 : ")
l1=int(input("Enter length : "))
b1=int(input("Enter breadth : "))
s1= Rectangle(l1,b1)
print("Area of rectangle 1",s1.area())

print("Enter parameters of rectangle 2 : ")
l2=int(input("Enter length : "))
b2=int(input("Enter breadth : "))
s2= Rectangle(l2,b2)
print("Area of rectangle 2",s2.area())

if(s1<s2):
```

```python
        print("Area of rectangle 2 is greater")
else:
        print("Area of rectangle 1 is greater")
```

OUTPUT

```
Enter parameters of rectangle 1 :
Enter length : 5
Enter breadth : 10
Area of rectangle 1 50

Enter parameters of rectangle 2 :
Enter length : 15
Enter breadth : 20
Area of rectangle 2 300

Area of rectangle 2 is greater
```

4. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 times.

PROGRAM

```python
class Time:
    def __init__(self,h,m,s):
        self.h=h
        self.m=m
        self.s=s
    def display(self):
        print(self.h,"hr ",self.m,"min ",self.s,"sec")
    def __add__(self, t):
        res = Time(0,0,0)
        res.h = self.h+t.h
        res.m = self.m+t.m
        res.s = self.s+t.s

        res.m=res.m+res.s//60
        res.s=res.s%60
        res.h=res.h+res.m//60
        res.m=res.m%60

        return res

h1, m1, s1 = input("Enter Time 1(H:M:S) : ").split()
h2, m2, s2 = input("Enter Time 2(H:M:S) : ").split()
t1=Time(int(h1), int(m1), int(s1))
t2=Time(int(h2), int(m2), int(s2))
```

```python
t3=t1+t2

print("Time 1 : ", end="")
t1.display()
print("Time 2 : ", end="")
t2.display()
print("Time 3 : ", end="")
t3.display()
```

OUTPUT

```
Enter Time 1(H:M:S) : 15 25 30
Enter Time 2(H:M:S) : 05 15 30

Time 1 : 15 hr  25 min  30 sec
Time 2 : 5 hr  15 min  30 sec
Time 3 : 20 hr  41 min  0 sec
```

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

```python
class Publisher:
    def __init__(self, name):
        self.name=name
    def display():
        pass

class Book(Publisher):
    def __init__(self, name, title, author):
        self.title=title
        self.author=author
        super().__init__(name)
    def display():
        pass

class Python(Book):
    def __init__(self, title, author, name, price, no_of_pages):
        self.price=price
        self.no_of_pages=no_of_pages
        super().__init__(name, title, author)
    def display(self):
        print("Book Details")
```

```python
        print("Title  : ",self.title)
print("Author  : ",self.author)
print("Publisher  : ",self.name)
print("Price  : ",self.price)
print("No of pages  : ",self.no_of_pages)

title, author, name, price, no_of_pages = input("Enter Book title, author
name, Publisher, Price and No of pages : ").split(',')
s= Python(title, author, name, int(price), int(no_of_pages))
s.display()
```

OUTPUT

```
Enter Book title, author name, Publisher, Price and No of pages :
Taming Python By Programming,Jeeva Jose,Kalyani
Publications,265,340

Book Details
Title  :  Taming Python By Programming
Author  :  Jeeva Jose
Publisher  :  Kalyani Publications
Price  :  265
No of pages  :  340
```