COURSE OUTCOME-4

DATE: 03/12/2024

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,length,breadth):
                self.length=length
                self.breadth=breadth
        def area(self):
                return self.length*self.breadth
        def perimeter(self):
                return 2*(self.length+self.breadth)
len1=int(input("Enter length of rectangle1:"))
bread1=int(input("Enter breadth of rectangle 1:"))
len2=int(input("Enter length of rectangle1:"))
bread2=int(input("Enter breadth of rectangle 1:"))
rect1=Rectangle(len1,bread1)
rect2=Rectangle(len2,bread2)

if rect1.area()>rect2.area():
        print("Area of Rectangle1 is greater than Rectangle2")
elif rect1.area()<rect2.area():
        print("Area of Rectangle2 is greater than Rectangle1")
else:
        print("Area of two rectangles is Equal")
```

OUTPUT

```
Enter length of rectangle1:20
Enter breadth of rectangle 1:10
Enter length of rectangle1:15
Enter breadth of rectangle 1:4
Area of Rectangle1 is greater than Rectangle2


Enter length of rectangle1:12
Enter breadth of rectangle 1:5
Enter length of rectangle1:13
Enter breadth of rectangle 1:6
Area of Rectangle2 is greater than Rectangle1
```

DATE: 03/12/2024

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

```
class Account:
        def __init__(self,accno,aname,acctype,balance):
                self.accno=accno
                self.aname=aname
                self.acctype=acctype
                self.balance=balance

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

        def withdraw(self,amt):
                if amt>self.balance:
                        print("Insufficient Balance")
                else:
                        self.balance-=amt
                        print("Successfully withdrawn ",amt)

        def viewDetails(self):
                print("\nAccount Number  : ",self.accno)
                print("Account Name    : ",self.aname)
                print("Account Type    : ",self.acctype)
                print("Account Balance : Rs.",self.balance,"\n")

accno=int(input("Enter account number :"))
aname=input("Enter account name :")
acctype=input("Enter account type:")
balance=int(input("Enter account balance :"))
c1=Account(accno,aname,acctype,balance)
while True:
        print("----------MENU----------\n1. Deposit\n2. Withdraw\n3. Currect
Balance\n
         4. 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("\nCurrent Balance : Rs.",c1.balance,"\n")

    elif ch==4:
        c1.viewDetails()
    elif ch==5:
       break
```

OUTPUT

```
Enter account number :446739
Enter account name :Thomas
Enter account type:fixed
Enter account balance :120000
----------MENU------------
1. Deposit
2. Withdraw
3. Currect Balance
4. View Details
5. Exit
Enter your choice :1
Enter the amount to be deposited:20000
Successfully Deposited  20000
----------MENU------------
1. Deposit
2. Withdraw
3. Currect Balance
4. View Details
5. Exit
Enter your choice :3

Current Balance : Rs. 140000

----------MENU------------
1. Deposit
2. Withdraw
3. Currect Balance
4. View Details
5. Exit
Enter your choice :2
Enter the amount to be withdrawn:10000
Successfully withdrawn  10000
----------MENU------------
1. Deposit
2. Withdraw
```

3. Currect Balance
4. View Details
5. Exit
Enter your choice :3

Current Balance : Rs. 130000

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

Account Number : 446739
Account Name   : Thomas
Account Type   : fixed
Account Balance : Rs. 130000

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

DATE: 05/12/2024

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,length,width):
                self.length=length
                self.width=width

        def area(self):
                return self.length*self.width

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

l1=int(input("Enter the length of Rectangle1:"))
w1=int(input("Enter the width of Rectangle1:"))
l2=int(input("Enter the length of Rectangle2:"))
w2=int(input("Enter the width of Rectangle1:"))

rect1=Rectangle(l1,w1)
rect2=Rectangle(l2,w2)
if rect1<rect2:
        print("Area of Rectangle1 is smaller than Area of Rectangle2")
elif(rect1>rect2):
        print("Area of Rectangle1 is larger than Area of Rectangle2")
else:
        print("Both Rectangles have same Area")
```

OUTPUT

```
Enter the length of Rectangle1:12
Enter the width of Rectangle1:5
Enter the length of Rectangle2:15
Enter the width of Rectangle1:6
Area of  Rectangle1  is  smaller  than  Area of Rectangle2



Enter the length of Rectangle1:20
Enter the width of Rectangle1:14
Enter the length of Rectangle2:11
Enter the width of Rectangle1:8
Area of Rectangle1 is larger than Area of Rectangle2
```

DATE: 05/12/2024

4. 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 displayTime(self):
                print(self.hour,"hr:",self.minute,"min:",self.second,"sec");

        def __add__(self,other):
                sum_seconds=self.second+other.second
                sum_minutes=self.minute+other.minute+(sum_seconds//60)
                sum_hours=self.hour+other.hour+(sum_minutes//60)
                sum_seconds=sum_seconds%60
                sum_minutes=sum_minutes%60
                return Time(sum_hours,sum_minutes,sum_seconds)


h1=int(input("Enter hour1:"))
m1=int(input("Enter minute1:"))
s1=int(input("Enter second1:"))
h2=int(input("\nEnter hour2:"))
m2=int(input("Enter minutes2:"))
s2=int(input("Enter seconds2:"))
t1 = Time(h1,m1,s1);
t2 = Time(h2,m2,s2);
t3 = t1 + t2
print("\nTime 1: ", end="")
t1.displayTime()
print("Time 2: ", end="")
t2.displayTime()
print("Sum of Time1 and Time2 is ", end="")
t3.displayTime()
```

OUTPUT

```
Enter hour1:12
Enter minute1:45
Enter second1:32
```

Enter hour2:3
Enter minutes2:27
Enter seconds2:12

Time 1: 12 hr: 45 min: 32 sec
Time 2: 3 hr: 27 min: 12 sec
Sum of Time1 and Time2 is 16 hr: 12 min: 44 sec

Enter hour1:2
Enter minute1:39
Enter second1:30

Enter hour2:1
Enter minutes2:20
Enter seconds2:30

Time 1: 2 hr: 39 min: 30 sec
Time 2: 1 hr: 20 min: 30 sec
Sum of Time1 and Time2 is 4 hr: 0 min: 0 sec

DATE: 05/12/2024

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,name):
                self.name=name
        def display():
                pass

class Book(Publisher):
        def __init__(self,name,title,author):
                super().__init__(name)   #invoking the base class constructor
                self.title=title
                self.author=author
        def display():
                pass
class Python(Book):
        def __init__(self,name,title,author,price,nopages):
                super().__init__(name,title,author)
                self.price=price
                self.nopages=nopages
        def display(self):
                print("\n-----------Book Details-----------")
                print("Title        : ",self.title)
                print("Name         : ",self.name)
                print("Author       : ",self.author)
                print("Price        : ",self.price)
                print("No. of Pages : ",self.nopages)

name=input("Enter Name :")
title=input("Enter Title:")
author=input("Enter Author:")
price=int(input("Enter Price:"))
nopages=int(input("Enter number of pages:"))
b=Python(name,title,author,price,nopages)
b.display()
```

OUTPUT

```
Enter Name :Rowling
Enter Title:Harry Potter
Enter Author:J K Rowling
Enter Price:650
Enter number of pages:328

----------Book Details----------
Title       :  Harry Potter
Name        :  Rowling
Author      :  J K Rowling
Price       :  650
No. of Pages :  328


Enter Name :Bronte
Enter Title:Wuthering Heights
Enter Author:Emily Bronte
Enter Price:450
Enter number of pages:467

----------Book Details----------
Title       :  Wuthering Heights
Name        :  Bronte
Author      :  Emily Bronte
Price       :  450
No. of Pages :  467
```

COURSE OUTCOME-5

DATE: 26/11/2024

1. **Write a Python program to read a file line by line and store it into a list.**

   PROGRAM

   file1.txt

   MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE VARIKOLI
   ERNAKULAM DISTRICT
   FATHIMA NAZRIN K A
   MCA

   file1.py

```
file=open("file1.txt","r")
l=[i.split() for i in open("file1.txt")]
print(l)
file.close()
```

   OUTPUT

   [['MUTHOOT', 'INSTITUTE', 'OF', 'TECHNOLOGY', 'AND', 'SCIENCE', 'VARIKOLI'], ['ERNAKULAM', 'DISTRICT'], ['FATHIMA',’NAZRIN’, 'K', 'A'], ['MCA']]

DATE: 26/11/2024

2. Python program to copy odd lines of one file to other

PROGRAM

file1.txt

```
MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE VARIKOLI
ERNAKULAM DISTRICT
FATHIMA NAZRIN K A
MCA
```

file2.py

```
f=open("file1.txt","r")
g=open("file2.txt","w")
print("Odd lines of File1 which are in File2:")
lno=1
for line in f:

  if lno%2!=0:
     g.write(line)
  lno=lno+1

g.close()
g=open("file2.txt","r")
print(g.read())
g.close()
f.close()
```

OUTPUT

```
Odd lines of File1 which are in File2:
MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE VARIKOLI
FATHIMA NAZRIN K A
```

DATE: 28/11/2024

3. Write a program to read each row from the CSV file and print a list of strings

PROGRAM

student.csv

```
Rollno,Name,Age,Course
101,Nazrin,21,MCA
102,Farhana,21,MCA
103,Safrin,21,MCA
104,Ferteena,21,MCA
```

CSV1.py

```
import csv
with open("student.csv",mode="r") as f:
        csvr=csv.reader(f)
        for row in csvr:
                print(row)
```

OUTPUT

```
['Rollno', 'Name', 'Age', 'Course']
['101', 'Nazrin', '21', 'MCA']
['102', 'Farhana', '21', 'MCA']
['103', 'Safrin', '21', 'MCA']
['104', 'Ferteena', '21', 'MCA']
```

DATE: 28/11/2024

4. Write a Python program to read specific columns of a given CSV file and print the content of the columns.

PROGRAM

student.csv

```
Rollno,Name,Age,Course
101,Nazrin,21,MCA
102,Farhana,21,MCA
103,Safrin,21,MCA
104,Ferteena,21,MCA
```

CSV2.py

```
import csv
with open("student.csv",mode="r") as f:
        csvr=csv.reader(f)
        print(" CSV File ")
        for row in csvr:
                print(row)
f.close()
f=open("student.csv","r")
col=csv.reader(f)
print("\nSpecific columns from CSV file")
print("----------------------------------------")
for i in col:
        print(i[1],i[3])
f.close()
```

OUTPUT

```
CSV File
['Rollno', 'Name', 'Age', 'Course']
['101', 'Nazrin', '21', 'MCA']
['102', 'Farhana', '21', 'MCA']
['103', 'Safrin', '21', 'MCA']
['104', 'Ferteena', '21', 'MCA']

Specific columns from CSV file
---------------------------------------
Name Course
Nazrin MCA
Farhana MCA
Safrin MCA
Ferteena MCA
```

DATE: 28/11/2024

5. Write a Python program to write a Python dictionary to a csv file. After writing the CSV file, read the CSV file and display the content.

PROGRAM

CSV3.py

```
import csv
mydict=[{'branch':'COE','cgpa':'9.0','name':'Hari','year':'2'},
        {'branch':'IT','cgpa':'8.9','name':'Sam','year':'2'},
        {'branch':'SE','cgpa':'9.2','name':'Rohan','year':'3'},
        {'branch':'COE','cgpa':'9.5','name':'Neha','year':'2'},
        {'branch':'IT','cgpa':'8.8','name':'Aswin','year':'1'},
        {'branch':'SE','cgpa':'8.6','name':'Teena','year':'1'}]
fields=['name','branch','year','cgpa']
filename="records.csv"
with open(filename,"w") as f:
        writer=csv.DictWriter(f,fieldnames=fields)
        writer.writeheader()
        writer.writerows(mydict)
f.close()
with open("records.csv","r") as f:
        row=csv.reader(f)
        print("Contents in the created CSV file:")
        for i in row:
                print(i)
f.close()
```

OUTPUT

records.py

```
name,branch,year,cgpa
Hari,COE,2,9.0
Sam,IT,2,8.9
Rohan,SE,3,9.2
Neha,COE,2,9.5
Aswin,IT,1,8.8
Teena,SE,1,8.6
```

```
Contents in the created CSV file:
['name', 'branch', 'year', 'cgpa']
['Hari', 'COE', '2', '9.0']
['Sam', 'IT', '2', '8.9']
['Rohan', 'SE', '3', '9.2']
['Neha', 'COE', '2', '9.5']
['Aswin', 'IT', '1', '8.8']
['Teena', 'SE', '1', '8.6']
```