

COURSE OUTCOME 4

DATE: 3/12/2024

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)
l1=int(input("enter length 1:"))
b1=int(input("enter breadth 1:"))
r1=Rectangle(l1,b1)
print("area is:",r1.area())
print("perimeter is:",r1.perimeter())
l2=int(input("enter length 2:"))
b2=int(input("enter breadth 2:"))
r2=Rectangle(l2,b2)
print("area is:",r2.area())
print("perimeter is:",r2.perimeter())

if r1.area()>r2.area():
    print("rectangle 1 has bigger area")
elif r1.area()<r2.area():
    print("rectangle 2 has bigger area")
else:
    print("both rectangle have same area")
```

OUTPUT

enter length 1:4
enter breadth 1:5
area is: 20
perimeter is: 18
enter length 2:6
enter breadth 2:4
area is: 24
perimeter is: 20
rectangle 2 has bigger area

enter length 1:6
enter breadth 1:4
area is: 24
perimeter is: 20
enter length 2:5
enter breadth 2:2
area is: 10
perimeter is: 14
rectangle 1 has bigger area

DATE: 5/12/2024

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,ano,aname,atype,abalance):
        self.ano=ano
        self.aname=aname
        self.atype=atype
        self.abalance=abalance
    def deposit(self,amt):
        if amt>0:
            self.abalance=self.abalance+amt
            print("successfully deposited ",amt,"RS")
            print("your current balance now is: ",self.abalance,"RS")
        else:
            print("invalid amount")
    def withdraw(self,amt):
        if amt>self.abalance:
            print("insufficient balance")
        else:
            print("successfully withdrawn ",amt,"RS")
            self.abalance=self.abalance-amt
            print("your current balance now is: ",self.abalance,"RS")
    def viewdetails(self):
        print("Account no: ",self.ano)
        print("Name: ",self.aname)
        print("Account Type: ",self.atype)
        print("Account Balance: ",self.abalance,"RS")
ano=int(input("enter acct no: "))
aname=input("enter name: ")
atype=input("enter acct type: ")
abalance=int(input("enter acct balace: "))

c1=account(ano,aname,atype,abalance)
while True:
```

```

print("Menu\n1.Deposit\n2.Withdraw\n3.Current balance\n4.View Details\n5.Exit")
ch=int(input("enter your choice: "))
if ch==1:
    amt=int(input("enter amount to be deposited: "))
    c1.deposit(amt)
elif ch==2:
    amt=int(input("enter amount to be withdrawn: "))
    c1.withdraw(amt)
elif ch==3:
    print("current balance= ",c1.abalance,"RS")
elif ch==4:
    c1.viewdetails()
elif ch==5:
    print("User Exit")
    break

```

OUTPUT

```

enter acct no: 1111
enter name: ABHIJITH
enter acct type: current
enter acct balace: 25000
Menu
1.Deposit
2.Withdraw
3.Current balance
4.View Details
5.Exit
enter your choice: 2
enter amount to be withdrawn: 5000
successfully withdrawn 5000RS
your current balance now is: 20000 RS
Menu
1.Deposit
2.Withdraw
3.Current balance
4.View Details
5.Exit
enter your choice: 1
enter amount to be deposited: 10000

```

successfully deposited 10000 RS
your current balance now is: 30000 RS

Menu

- 1.Deposit
- 2.Withdraw
- 3.Current balance
- 4.View Details
- 5.Exit

enter your choice: 2

enter amount to be withdrawn: 12000

successfully withdrawn 2000 RS

your current balance now is: 18000 RS

Menu

- 1.Deposit
- 2.Withdraw
- 3.Current balance
- 4.View Details
- 5.Exit

enter your choice: 3

current balance= 18000 RS

Menu

- 1.Deposit
- 2.Withdraw
- 3.Current balance
- 4.View Details
- 5.Exit

enter your choice: 4

Account no: 1111

Name: ABHIJITH

Account Type: current

Account Balance: 18000 RS

Menu

- 1.Deposit
- 2.Withdraw
- 3.Current balance
- 4.View Details
- 5.Exit

enter your choice: 5

User Exit

DATE: 5/12/2024

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 length 1:"))
b1=int(input("enter width 1:"))
rectangle1=Rectangle(l1,b1)
l2=int(input("enter length 2:"))
b2=int(input("enter width 2:"))
rectangle2=Rectangle(l2,b2)
if rectangle1 < rectangle2:
    print("area of rectangle1 less than area of rectangle2")
elif rectangle1 > rectangle2:
    print("area of rectangle1 larger than area of rectangle2")
else:
    print("both rectangle have same area")
```

OUTPUT

enter length 1:4
enter width 1:5
enter length 2:8
enter width 2:6
area of rectangle1 less than area of rectangle2

enter length 1:9
enter width 1:5
enter length 2:4
enter width 2:6
area of rectangle1 larger than area of rectangle2

DATE: 5/12/2024

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 __add__(self, other):
        self.hour = self.hour + other.hour
        self.second = self.second + other.second
        if self.second > 60:
            self.minute = self.minute + 1
            self.second = self.second - 60
        self.minute = self.minute + other.minute
        if self.minute > 60:
            self.hour = self.hour + 1
            self.minute = self.minute - 60
        return self
    def display(self):
        print("Hour:", self.hour)
        print("Minute:", self.minute)
        print("Second:", self.second)
h1 = int(input("enter hour 1:"))
m1 = int(input("enter minute 1:"))
s1 = int(input("enter second 1:"))
time1 = Time(h1, m1, s1)
h2 = int(input("enter hour 2:"))
m2 = int(input("enter minute 2:"))
s2 = int(input("enter second 2:"))
time2 = Time(h2, m2, s2)
time3 = Time(0, 0, 0)
time3 = time1 + time2
time3.display()
```


OUTPUT

enter hour 1:5
enter minute 1:20
enter second 1:20
enter hour 2:4
enter minute 2:30
enter second 2:30
Hour: 9
Minute: 50
Second: 50

enter hour 1:3
enter minute 1:40
enter second 1:30
enter hour 2:4
enter minute 2:30
enter second 2:50
Hour: 8
Minute: 11
Second: 20

DATE: 6/12/2024

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 base class constructor here
        self.title=title
        self.author=author
    def display():
        pass
class Python(Book):
    def __init__(self,name,title,author,price,npage):
        super().__init__(name,title,author)
        self.price=price
        self.npage=npage
    def display(self):
        print("book details")
        print("Title: ",self.title)
        print("Publisher Name: ",self.name)
        print("Author: ",self.author)
        print("Price: ",self.price)
        print("No Of Pages: ",self.npage)

title=input("enter book title: ")
name=input("enter publisher name: ")
author=input("enter author name: ")
price=int(input("enter book price: "))
npage=int(input("enter no of pages: "))
```

```
b=Python(title,name,author,price,npage)
b.display()
```

OUTPUT

```
enter book title: To Kill a Mockingbird
enter publisher name: J.B. Lippincott & Co
enter author name: Harper Lee
enter book price: 200
enter no of pages: 40
book details
Title: To Kill a Mockingbird
Publisher Name: J.B. Lippincott & Co
Author: Harper Lee
Price: 200
No Of Pages: 40
```

```
enter book title: 1984
enter publisher name: Secker & Warburg
enter author name: George Orwell
enter book price: 300
enter no of pages: 60
book details
Title: 1984
Publisher Name: Secker & Warburg
Author: George Orwell
Price: 300
No Of Pages: 60
```

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

```
file=open("text2.txt","r")  
l=[i.split() for i in open("text2.txt")]  
print(l)
```

text2.txt

Muthoot Institute Of Technology And Science
Varikoli Puthencurez
KTU

OUTPUT

```
[['Muthoot', 'Institute', 'Of', 'Technology', 'And', 'Science'], ['Varikoli', 'Puthencurez'],  
['KTU']]
```

DATE: 26/11/2024

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

PROGRAM

```
f = open("text2.txt", "r")
print(f.read())
f.close()
f = open("text2.txt", "r")
g = open("text3.txt", "w")
h = open("text5.txt", "w")
i=1
for x in f:
    if i % 2!=0:
        g.write(x)
    else:
        h.write(x)
    i=i+1
g.close()
h.close()
f.close()
print("odd lines")
g = open("text3.txt", "r")
print(g.read())
g.close()
print("even lines")
h = open("text5.txt", "r")
print(h.read())
h.close()
```

text2.txt

Muthoot Institute Of Technology And Science
Varikoli Puthencurez
KTU

OUTPUT

Muthoot Institute Of Technology And Science
Varikoli Puthencurez
KTU

odd lines
Muthoot Institute Of Technology And Science
KTU

even lines
Varikoli Puthencurez

//new text files

text3.txt

Muthoot Institute Of Technology And Science
KTU

text5.txt

Varikoli Puthencurez

DATE: 28/11/2024

3. Write a Python program to read each row from a given csv file and print a list of Strings.

PROGRAM

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

student.csv

rollno	name	age	course
101	Abhijith	23	mca
102	basil	22	mca
103	jerin	22	mca
104	eldho	25	mca
105	thomas	23	mca
106	vishnu	23	mca

OUTPUT

```
all rows  
['rollno', 'name', 'age', 'course']  
['101', 'Abhijith', '23', 'bca']  
['102', 'basil', '22', 'bca']  
['103', 'jerin', '22', 'bca']  
['104', 'eldho', '25', 'bca']  
['105', 'thomas', '23', 'bca']  
['106', 'vishnu', '23', 'bca']
```

DATE: 28/11/2024

4. Write a Python program to read each row from a given csv file and print a list of Strings.

PROGRAM

```
with open("student.csv",mode="r") as file:
    csvr=csv.reader(file)
    print("all rows")
    for row in csvr:
        print(row)
    print()
    a=int(input("enter column:"))
    file.seek(0)
    print("particular column")
    for x in csvr:
        print(x[a])
```

student.csv

rollno	name	age	course
101	Abhijith	23	mca
102	basil	22	mca
103	jerin	22	mca
104	eldho	25	mca
105	thomas	23	mca
106	vishnu	23	mca

OUTPUT

all rows

```
['rollno', 'name', 'age', 'course']
```

```
['101', 'Abhijith', '23', ' mca ']
```

```
['102', 'basil', '22', ' mca ']
```

```
['103', 'jerin', '22', ' mca ']
```

```
['104', 'eldho', '25', ' mca ']
```

```
['105', 'thomas', '23', ' mca ']
```

```
['106', 'vishnu', '23', ' mca ']
```

enter column:1

particular column

name

Abhijith

basil

jerin

eldho

thomas

vishnu

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

```
dicteg=[
    {"roll no":1,"name":"allen","branch":"mca","age":21},
    {"roll no":2,"name":"abhijith","branch":"mca","age":23},
    {"roll no":3,"name":"adwaith","branch":"mca","age":22},
    {"roll no":4,"name":"vishnu","branch":"mca","age":23},
    {"roll no":5,"name":"thomas","branch":"mca","age":23},
    {"roll no":6,"name":"joyal","branch":"mca","age":22}]
field=["roll no","name","branch","age"]
filename="dictegfile.csv"
with open(filename,mode="w") as file:
    writer=csv.DictWriter(file,fieldnames=field)
    writer.writeheader()
    writer.writerows(dicteg)

with open(filename,mode="r") as file:
    csvr=csv.reader(file)
    for x in csvr:
        print(x)
```

dictegfile.csv

roll no	name	branch	age
1	allen	mca	21
2	abhijith	mca	23
3	adwaith	mca	22
4	vishnu	mca	23
5	thomas	mca	23
6	joyal	mca	22

OUTPUT

```
['roll no', 'name', 'branch', 'age']  
['1', 'allen', 'mca', '21']  
['2', 'abhijith', 'mca', '23']  
['3', 'adwaith', 'mca', '22']  
['4', 'vishnu', 'mca', '23']  
['5', 'thomas', 'mca', '23']  
['6', 'joyal', 'mca', '22']
```