FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

PROGRAMMING LAB

……..

LABORATORY RECORD

Name: DEEPA P D

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: SEMESTER -1 A Roll No: 50

FEDERAL INSTITUTE OF

SCIENCE AND TECHNOLOGY

(FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name :DEEPA P D

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 **Roll No: 50**

University Exam.Reg. No: FIT21MCA-2050

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to Kerala Technological University in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by **DEEPA P D** in the **PROGRAMMING** Laboratory of the Federal Institute of Science and Technology during the academic year 2020-2021.

Signature of Staff in Charge — Signature of H.O.D

Name: — Name:

Date:

Date of University practical examination ……

Signature of Internal Examiner — Signature of External Examiner

14	**17/11/21**	Print out all colors from color-list 1 not contained in color-list2.	**20**	
15	**2/12/21**	Create a single string separated with space from two strings by swapping the character at position 1	**21**	
16	**2/12/21**	Sort dictionary in ascending and descending order.	**22**	
17	**2/12/21**	Merge two dictionaries.	**23**	
18	**25/11/21**	Find gcd of 2 numbers.	**23**	
19	**17/11/21**	From a list of integers, create a list removing even numbers.	**24**	
20	**28/10/21**	Program to find the factorial of a number	**25**	
21	**28/10/21**	Generate Fibonacci series of N terms	**26**	
22	**2/12/21**	Find the sum of all items in a list	**27**	
23	**2/12/21**	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	**28**	
24	**2/12/21**	Display the given pyramid with step number accepted from user	**29**	
25	**2/12/21**	Count the number of characters (characterfrequency) in a string.	**30**	
26	**2/12/21**	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’	**31**	
27	**2/12/21**	Accept a list of words and return length of longest word.	**32**	
28	**2/12/21**	Construct following pattern using nested loop	**33**	
29	**2/12/21**	Generate all factors of a number.	**34**	
30	**20/01/22**	Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements.	**35-38**	
31	**9/12/21**	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	**39-40**	

32	**9/12/21**	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.	**41-42**	
33	**16/12/21**	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	**43**	
34	**13/01/22**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	**44**	
35	**20/01/22**	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 ages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.	**45-46**	
36	**27/01/22**	Write a Python program to read a file line by line and store it into a list.	**47**	
37	**27/01/22**	Write a Python program to read each row from a given csv file and print a list of strings.	**48**	

Course Outcome 1(CO1)

PROGRAM 1

AIM

Display future leap years from current leap year to a final year entered by user.

Input

```
#leap
print ("Enter last year")
endYear = int(input())
print ("List of leap years:")
for year in range(2021, endYear+1):
    if (year % 4 == 0 ) and (year % 100 != 0) or (year % 400 == 0):
        print (year)
```

Output

```
stud@debian:~/deepa$ python3 leap.py
Enter last year
2030
List of leap years:
2024
2028
stud@debian:~/deepa$
```

PROGRAM 2

AIM

List comprehensions:
(a) Generate positive list of numbers from a given list of integers.

Input

```
#positive no in list
list1=[0,-1,-2,3,4,1,2,5]
    for num in list1:
        if (num>=0):
    print(num)
```

Output

```
stud@debian:~/deepa/d$ python3 poli3aco1.py
0
3
4
1
2
5
```

AIM

(b)Square of N numbers

Input

```
#Square of n numbers
list1=[3,56,6]
list2=[]
print("The squares of the given numbers are:")
for i in list1:
    s=(i*i) print(s)
```

Output

```
stud@debian:~/deepa/d$ python3 sq3bco1.py
The squares of the given numbers are:
9
3136
36
```

AIM

(c)Form a list of vowels selected from a given word

Input

```
#vowel in string
word=input('Enter the string:')
vowel=['a','e','i','o','u']
list1=[]
for d in word:
if(d in vowel and d not in list1):
        list1.append(d)
print('vowels are:',list1)
```

Output

```
stud@debian:~/deepa/d$ python3 vo3cco1.py
Enter the string:welcome
vowels are: ['e', 'o'] _
```

AIM

(d)List ordinal value of each element of a word

Input

```
#ordinal value
word=input('Enter the string:')
print([ord(x) for x in word])
```

Output

```
stud@debian:~/deepa/d$ python3 ord.py
Enter the string:welcome
[119, 101, 108, 99, 111, 109, 101]
```

PROGRAM 3

AIM

Count the occurrences of each word in a line of text.

Input

```
list1=[]
list2=[]
x=input("Enter a string:")
for i in x.split(" "):
        list1.append(i)
        if i not in list2:
                list2.append(i)
for i in list2:
        print(i,"\t",list1.count(i))
```

Output

```
PS C:\Users\deepa\Downloads\python> python3 4co1.py
Enter a string:Anu Achu Anu Ammu
Anu      2
Achu     1
Ammu     1
PS C:\Users\deepa\Downloads\python>
```

PROGRAM 4

AIM

Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

Input

```
#store over for values>100
w=int(input('Enter the limit:'))
list1=[]
for i in range(0,w):
        value=int(input('Enter the values:'))
        if(value>100):
                list1.append('over')
        else:
                list1.append(value)
print(list1)
```

Output

```
stud@debian:~/deepa/d$ python3 over.py
Enter the limit:4
Enter the values:12
Enter the values:102
Enter the values:201
Enter the values:45
[12, 'over', 'over', 45]
```

PROGRAM 5

AIM

Store a list of first names. Count the occurrences of ‘a’ within the list.

Input

```
#counta word=['anna','anu']
r=0
for d in word:
        for c in d:
                if (c=='a'):
                        r=r+1
print(r)
```

Output

```
PS C:\Users\deepa\Downloads\python> python3 counta.py
Occurance of a in the given list is 3
PS C:\Users\deepa\Downloads\python>
```

PROGRAM 6

AIM

Enter 2 list of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both.

Input

```
#lensame l1=[1,2,3,4,5]
l2=[6,3,21,6,]
p=len(l1)
q=len(l2)
if(p==q):
        print("The length of two lists are same")
else:
        print("The length of lists are not same")
s=0
p=0
for i in l1:
        s=s+i
print("Sum of list1 is",s)
for r in l2:
        p=p+r
print("Sum of list2 is",p)
if(s==p):
        print("Sum of elements in two lists are same")
else:
        print("Sum of elements in two lists are not same")
l3=[]
f=0
for i in l1:
        if i in l2:
                l3.append(i)
                f=f+1
print(l3)
if(f==0):
        print("no element is same")
```

Output

```
PS C:\Users\deepa\Downloads\python> python3 7co1.py
The length of lists are not same
Sum of list1 is 15
Sum of list2 is 36
Sum of elements in two lists are not same
values that occur in both list: [3]
PS C:\Users\deepa\Downloads\python>
```

PROGRAM 7

AIM

Get a string from an input string where all occurrences of first character replaced with '$', except first character.

Input

```
#character replace
str1=input("Enter a string:")
print("Original string:",str1)
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1: ]
print("Replaced string: ",str1)
```

Output

```
stud@debian:~/deepa$ python3 8co1.py
Enter a string:onion
Original string: onion
Replaced string:  oni$n
```

PROGRAM 8

AIM

Create a string from given string where first and last characters exchanged.

Input

```
#First and last character exchange
str=input("Enter a string:")
char=str[0]
char1=str[-1]
n=len(str)
ns=char1+str[1:n-1]+char print(ns)
```

Output

```
stud@debian:~/deepa/d$ python3 9co1.py
Enter a string:python
nythop
```

PROGRAM 9

AIM

Accept the radius from user and find area of circle.

Input

```
#Area of the circle

x=int(input('Enter the radius:'))

A=3.14*x*x

print("Area of the circle is",A)
```

Output

```
stud@debian:~/deepa$ python3 area.py
Enter the radius:2
Area of the circle is 12.56
```

PROGRAM 10

AIM

Find biggest of 3 numbers entered.

Input

```
#biggest of 3 nos
a=int(input('Enter the first number:'))
b=int(input('Enter the second number:'))
c=int(input('Enter the third number:'))
if a>b:
        if a>c:
                print(a)
        else:
                if(b>c):
                        print(b)
else:
        print(c)
```

Output

```
stud@debian:~/deepa$ python3 big.py
Enter the first number:2
Enter the second number:5
Enter the third number:6
Biggest of the three number is:
6
```

PROGRAM 11

AIM

Accept a file name from user and print extension of that.

Input

```
#extention of file
import os
a=input("Enter the file name\n")
print(os.path.splitext(a))
```

Output

```
stud@debian:~/deepa/d$ python3 12co1.py
Enter the file name:12co1.py
The extention of file 12co1.py is ('12co1', '.py')
```

PROGRAM 12

AIM

Create a list of colors from comma-separated color names entered by user. Display first and last colors.

Input

```
#first and last colours
l1=['Red','blue','white','yellow','Black']
print("First and last colours in the list are:")
print(l1[0],'and',l1[-1])
```

Output

```
stud@debian:~/deepa/d$ python3 13col.py
First and last colours in the list are:
Red and Black
```

PROGRAM 13

AIM

Accept an integer n and compute n+nn+nnn.

Input

```
#computing n+nn+nnn
x=int(input("enter the numbers"))
a=str(x)
b=a+a
c=a+a+a
d=x+int(b)+int(c)
print(d)
```

Output

```
stud@debian:~/deepa/d$ python3 16co1.py
Enter a number:2
246
```

PROGRAM 14

AIM

Print out all colors from color-list1 not contained in color-list2.

Input

```
#colours not in list2
l1=['red','blue','black']
l2=['red','white','pink']
l3=[]
for i in l1:
        if i not in l2:
                l3.append(i)
print('colours not in l2 is:\n',l3)
```

Output

```
stud@debian:~/deepa$ python3 15color.py
colours not in l2 is:
 ['blue', 'black']
```

PROGRAM 15
AIM

Create a single string separated with space from two strings by swapping the character at position 1.

Input

```
#swapping
str1=input("Enter first string:")
str2=input("Enter second string:")
str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]
print(str3)
```

Output

```
stud@debian:~/deepa$ python3 16coo1.py
Enter first string:code
Enter second string:Analysis
Aode cnalysis
```

PROGRAM 16

AIM

Sort dictionary in ascending and descending order.

Input

```
#ascending and descending order
d1={"annie":1,"carolin":3,"danic":2,"baachu":4}
l=list(d1.items())


print("orginal list is",l)
l.sort()
print("Ascending order is\n",l)
l=list(d1.items())
l.sort(reverse=True)
print("Desencding order is\n",l)
```

Output

```
stud@debian:~/deepa$ python3 17co1.py
orginal list is [('annie', 1), ('carolin', 3), ('danic', 2), ('baachu', 4)]
Ascending order is
 [('annie', 1), ('baachu', 4), ('carolin', 3), ('danic', 2)]
Desencding order is
 [('danic', 2), ('carolin', 3), ('baachu', 4), ('annie', 1)]
```

PROGRAM 17

AIM

Merge two dictionaries

Input

```
#Merge two dictionary
d1={'name': "deepa" , 'age': '21'}
d2={'qlfn':"pg"}
d1.update(d2)
print(d1)
```

Output

```
stud@debian:~/deepa/d$ python3 18co1.py
{'name:DEEPA', 'age:21', 'qfn:PG'}
```

PROGRAM 18

AIM

Find gcd of 2 numbers.

Input

```
#gcd of 2 nos
x=(int(input("Enter 1st number\n")))
y=(int(input("Enter 2nd number\n")))
z = min(x,y)
for i in range(1,z+1):
        if((x%i)==0 and (y%i==0)):
                gcd=i print("GCD is ",gcd)
```

Output

```
stud@debian:~/deepa$ python3 gcd.py
Enter the number:24
Enter the number:6
GCD is 6
```

PROGRAM 19

AIM

From a list of integers, create a list removing even numbers.

Input

```
#list removing even nos
l1=[1,2,3,4]
l3=[]
print("Even numbers in the given list are:")
for i in l1:
    if(i%2==0):
        print(i)
    else:

        l3.append(i)
print("Removing even numbers:",l3)
```

Output

```
stud@debian:~/deepa/d$ python3 20reeven.py
Even numbers in the given list are:
2
4
Removing even numbers: [1, 3]
```

Course Outcome 2(CO2):

PROGRAM 20

AIM

Program to find the factorial of a number.

Input

```
#co21factorial of a number
n=int(input('Enter the number:'))
fact=1
for i in range(1,n+1):
        fact=fact*i
print(fact)
```

Output

```
stud@debian:~/deepa/d$ python3 factorial.py
Enter the number:4
24
```

PROGRAM 21

AIM

Generate Fibonacci series of N terms.

Input

```
#co22fibanocciseries
n=int(input('Enter number of terms:'))
f1=0
f2=1
print(f1,f2)
for i in range(0,n):
    f3=f1+f2
    print(f3)
    f1=f2
    f2=f3
```

Output

```
stud@debian:~/deepa/d$ python3 fibonacci.py
Enter number of terms:4
0
1
1
2
3
5
```

PROGRAM 22

AIM

Find the sum of all items in a list.

Input

```
#sum of items in list
l1=[1,2,3,4]
l3=[]
s=0
p=0
for i in l1:
        s=s+i
print("Sum of list1 is",s)
```

Output

```
stud@debian:~/deepa/d$ python3 3co2.py
Sum of list1 is 10
```

PROGRAM 23

AIM

Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

Input

```
limit1=1000
limit2=9999
list1=[]
for i in range(limit1,limit2):
        j=i
        digit=[]
        while(i!=0):
                digit.append(i%10)
                i=int(i/10)
        count=0
        for n in digit:
                if n%2==0:
                        count=count+1
        if count==4:
                for k in range(31,100):
                        if((k**2)==j):
                                list1.append(j)
                                print(k)
print(list1)
```

Output

```
stud@debian:~/deepa$ python3 co2p4.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

PROGRAM 24

AIM

Display the given pyramid with step number accepted from the user.

Eg: N=4

```
1
2 4
3 6 9
4 8 12 16
```

Input

```
#pyramid with step no
n=int(input('enter the step number'))
for i in range(1,n+1):
    for j in range(1,i+1):
        s=i*j
        print(s,'\t',end="")
    print("\n")
```

Output

```
stud@debian:~/deepa/d$ python3 py5.py
Enter the step no:4
1
2       4
3       6       9
4       8       12      16
```

PROGRAM 25

AIM

Count the number of characters (character frequency) in a string.

Input

```
#character frequency
str=input("Enter a string:")
fnd=input("Enter character:")
cnt=0
fnd=fnd.lower()
str=str.lower()
for i in str:
        if i==fnd:
                cnt=cnt+1
print("Freq:→",cnt)
```

Output

```
stud@debian:~/deepa$ python3 6co2.py
Enter a string:India is our Nation
Enter character:i
Freq:-> 4
```

PROGRAM 26

AIM

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.

Input

```
#add ing
a=input("Enter a word\n")
l=len(a)
ll=a[l-3:l]
if(ll=="ing"):
        s=a+"ly"
else:
        s=a+"ing"
print (s)
```

Output

```
stud@debian:~/deepa$ python3 7co2.py
Enter a string:writing
writingly
stud@debian:~/deepa$ python3 7co2.py
Enter a string:write
writeing
```

PROGRAM 27

AIM

Accept a list of words and return length of longest word.

Input

```
#Length of the
longest word

lis=[]
n=int(input("Enter the
range:")) print("Enter the
words:")

for i in
range(0,n)
:

lis.append
(input(""))
longest=li
s[0]

for i in
  range(1,n):

  if(len(lis[i])>l
  en(longest)):

    longest=lis[i]

print("Length of longest word is",len(longest))
```

Output

```
stud@debian:~/deepa$ python3 8co2.py
Enter the range:4
Enter the words:
India
Sreelanka
Iran
America
Length of longest word is 9
```

PROGRAM 28

AIM

Construct following pattern using nested loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Input

```
#pattern star
for i in range(0,5):
        for j in range(0,i):
                print("*",end="")
        print("\n")
for i in range(5,0,-1):
        for j in range(i,0,-1):
                print("*",end="")
        print("\n")
```

Output

```
stud@debian:~/deepa/d$ python3 9co2.py
*

**

***

****

*****

***

**

*
```

PROGRAM 29

AIM

Generate all factors of a number.

Input

```
#All factors of a no
n=int(input("Enter a number:"))
print("Factors are")

for i in range(1,n+1):
        if(n%i==0):
                print(i)
```

Output

```
stud@debian:~/deepa$ python3 10co2.py
Enter a number:6
Factors are
1
2
3
6
```

Course Outcome 3(CO3):

PROGRAM 30

AIM

Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements.

Input

Graphics\circle.py

```
from math import pi

def area_circle(radius):
    return pi*radius*radius

def perimeter_circle(radius):
    return 2*pi*radius
```

Graphics\rectangle.py

```
def area_rec(length,width):
    return length*width

def perimeter_rec(length,width):
    return 2*(length+width)
```

Graphics\tdgraphics\cuboid.py

```
def area_cuboid(l,b,h):
        return 2*(l*h + b*h + l*b)

def volume_cuboid(l,b,h):
        return l*b*h
```

Graphics\tdgraphics\sphere.py

```
from math import pi

def area_sphere(radius):
        return 4*(pi*radius*radius)

def perimeter_sphere(radius):
        return 2*pi*radius
```

Graphics.py (driver code)

```
import graphics
from graphics import circle,rectangle
from graphics.tdgraphics import cuboid,sphere
from graphics.circle import *

print("Area of a circle with radius 20 is : ",circle.area_circle(20))
print("Permeter of a circle with radius 20 is ",circle.perimeter_circle(20))
print("\n")

print("Area of a Rectangle with length 20 and width 10 is :
",rectangle.area_rec(20,10))
print("Permeter of a Rectangle with length 20 and width 10 is :
",rectangle.perimeter_rec(20,10))
print("\n")

print("Area of a cuboid with length,width,height 8 is : ",cuboid.area_cuboid(8,8,8))
print("Volume of a cuboid with length,width,height 12 is :
",cuboid.volume_cuboid(12,12,12))
print("\n")

print("Area of a spere with radius 20 is : ",sphere.area_sphere(20))
print("Permeter of a spere with radius 20 is ",sphere.perimeter_sphere(20))
```

```
C:\Users\deepa>cd downloads

C:\Users\deepa\Downloads>cd python

C:\Users\deepa\Downloads\python>md graphics

C:\Users\deepa\Downloads\python>cd graphics

C:\Users\deepa\Downloads\python\graphics>notepad __init__.py

C:\Users\deepa\Downloads\python\graphics>notepad circle.py

C:\Users\deepa\Downloads\python\graphics>notepad rectangle.py

C:\Users\deepa\Downloads\python\graphics>md tdgraphics

C:\Users\deepa\Downloads\python\graphics>cd tdgraphics

C:\Users\deepa\Downloads\python\graphics\tdgraphics>notepad __init__.py

C:\Users\deepa\Downloads\python\graphics\tdgraphics>notepad cuboid.py

C:\Users\deepa\Downloads\python\graphics\tdgraphics>notepad sphere.py
```

Output

```
C:\Users\deepa\Downloads\python\graphics\tdgraphics>cd ..

C:\Users\deepa\Downloads\python\graphics>cd ..

C:\Users\deepa\Downloads\python>python3 graphics.py
Area of a circle with radius 20 is :  1256.6370614359173
Perimeter of a circle with radius 20 is  125.66370614359172


Area of a Rectangle with length 20 and width 10 is :  200
Perimeter of a Rectangle with length 20 and width 10 is :  60


Area of a cuboid with length,width and height 8 is :  384
Volume of a cuboid with length,width and height 12 is :  1728


Area of a sphere with radius 20 is :  5026.548245743669
Permieter of a sphere with radius 20 is  125.66370614359172

C:\Users\deepa\Downloads\python>
```

Course Outcome 4(CO4):

PROGRAM 31

AIM

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
         def area(self):
            return (self.l*self.b)
         def perimeter(self):
            return 2*(self.l+self.b)
p=int(input("Enter length of first rectangle:"))
q=int(input("Enter breadth of first rectangle:"))
r=int(input("Enter length of second rectangle:"))
s=int(input("Enter breadth of second rectangle:"))
r1=Rectangle(p,q)
r2=Rectangle(r,s)
x=r1.area()
y=r2.area()
z=r1.perimeter()
h=r2.perimeter()
if(x>y):
        print("Area of first rectangle is greater")
else:
        print("Area of second rectangle is greater")
print("Perimeter of first rectangle is",z)
print("Perimeter of second rectangle is",h)
```

Output

```
PS C:\Users\deepa\downloads> python3 1co4.py
Enter length of first rectangle:5
Enter breadth of first rectangle:4
Enter length of second rectangle:3
Enter breadth of second rectangle:2
Area of first rectangle is greater
Perimeter of first rectangle is 18
Perimeter of second rectangle is 10
```

PROGRAM 32

AIM

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:
    def __init__(self,acno,name,typeofac,balance):
        self.acno=acno
        self.name=name
        self.typeofac=typeofac
        self.balance=balance
    def withdraw(self,x):
        self.balance=self.balance-x
        print("Balance is:",self.balance)
    def deposit(self,y):
        self.balance=self.balance+y
        print("Balance is:",self.balance)

ac1=Bank(1,"Aiswarya","SB",10000)
ac2=Bank(2,"Krishnenthu","SB",20000)
p=int(input("Enter amount to withdraw:"))
q=int(input("Enter amount to deposit:"))
r=int(input("Enter amount to withdraw:"))
ac1.withdraw(p)
ac2.deposit(q)
ac1.deposit(r)
```

Output

```
PS C:\Users\deepa\downloads> python3 2co4.py
Enter amount to withdraw:1000
Enter amount to deposit:2000
Enter amount to withdraw:2000
Balance is: 9000
Balance is: 22000
Balance is: 11000
```

PROGRAM 33

AIM

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

Input

```
class Rectangle:
    def __init__(self,length,breadth):
        self.__length=length
        self.__breadth=breadth
    def area(self):
        a=self.__length*self.__breadth
        print("area",a)
        return a
    def perimeter(self):
        p=2*(self.__length+self.__breadth)
        print("perimeter",p)
    def __lt__(self,rr):
        if(self.__breadth*self.__length>rr.__breadth*rr.__length):
            return True
        else:
            return False
r1=Rectangle(5,7)
r2=Rectangle(4,6)
if(r1<r2):
    print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
```

Output

```
PS C:\Users\deepa\downloads> python3 3co4.py
Area of first rectangle is greater
```

PROGRAM 34

AIM

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,hr,min,sec):
                self.hr=hr
                self.min=min
                self.sec=sec
        def __add__(t1,t2):
                hr=t1.hr+t2.hr
                min=t1.min+t2.min
                sec=t1.sec+t2.sec
                print("The Sum of Two Times is",hr,":",min,":",sec)
t1=Time(2,30,46)
t2=Time(4,20,2)
t1+t2
```

Output

```
PS C:\Users\deepa\downloads> python3 4co4.py
The Sum of Two Times is 6 : 50 : 48
```

PROGRAM 35

AIM

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 ages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Input

```
class Publisher:
    def __init__(self,name):
        self.name=name
class Book(Publisher):
    def __init__(self,name,title,author):
        super().__init__(name)
        self.title=title
        self.author=author
class Python(Book):
    def __init__(self,name,title,author,price,no_of_pages):
        super().__init__(name,title,author)
        self.price=price
        self.no_of_pages=no_of_pages
    def display(self):
        print("Name:",self.name)
        print("Title:",self.title)
        print("Author:",self.author)
        print("Price:",self.price)
        print("No of pages:",self.no_of_pages)
p1=Python("Times publications","Python
Programming","Mr.James",480,210)
p1.display()
```

Output

```
PS C:\Users\deepa\downloads\python> python3 4c4.py
Name: Times publications
Title: Python Programming
Author: Mr.James
Price: 480
No of pages: 210
PS C:\Users\deepa\downloads\python>
```

Course Outcome 5(CO5):

PROGRAM 36

AIM

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

Input

```
f=open("data_file.txt","w")
f.write("India 10")
f.write("\n")
f.write("Australia 20")
f.close()
f=open("data_file.txt","r")
for x in f.readlines():
    print(x)
```

Output

```
----
C:\Users\deepa


PS C:\Users\deepa> cd downloads
PS C:\Users\deepa\downloads> cd python
PS C:\Users\deepa\downloads\python> python3 1co5.py
India 10

Australia 20
```

PROGRAM 37

AIM

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

Input

```
import csv
with open("profession.csv","r")as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
```

Output

```
PS C:\Users\deepa\downloads> cd python
PS C:\Users\deepa\downloads\python> python3 2co5.py
['NAME', 'AGE', 'PROFESSION']
['MANU', '30', 'MANAGER']
['RAMYA', '28', 'ACCOUNTANT']
['ANU', '25', 'PROFESSOR']
['ATHARV', '26', 'ENGINEER']
PS C:\Users\deepa\downloads\python>
```