FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ANJANA RAJEEV

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 24**

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

*This is to certify that this is a Bonafide record of the Practical work done by Anjana Rajeev in the **20MCA131 PROGRAMMING LAB** Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H O D

Name:

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

CONTENT

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
1		Display future leap years from current year to a final year entered by user.		
2		List comprehensions: (a) Generate positive list of numbers from a given list of integers (b) Square of N numbers (c) Form a list of vowels selected from a given word (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)		
3		Count the occurrences of each word in a line of text.		
4		Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.		
5		Store a list of first names. Count the occurrences of 'a' within the list		
6		Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both		
7		Get a string from an input string where all occurrences of first character replaced with '$', except first character. [eg: onion ->oni$n]		
8		Create a string from given string where first and last characters exchanged. [eg: python - >nythop]		
9		Accept the radius from user and find area of circle.		
10		Find biggest of 3 numbers entered.		
11		Accept a file name from user and print extension of that.		

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
12		Create a list of colors from comma-separated color names entered by user. Display first and last colors.		
13		Accept an integer n and compute n+nn+nnn.		
14		Print out all colors from color-list1 not contained in color-list2.		
15		Create a single string separated with space from two strings by swapping the character at position 1.		
16		Sort dictionary in ascending and descending order.		
17		Merge two dictionaries.		
18		Find gcd of 2 numbers.		
19		From a list of integers, create a list removing even numbers.		
20		Program to find the factorial of a number.		
21		Generate Fibonacci series of N terms.		
22		Find the sum of all items in a list		
23		Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.		
24		Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 8 12 16		
25		Count the number of characters (character frequency) in a string.		
26		Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’		

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
27		Accept a list of words and return length of longest word.		
28		Construct following pattern using nested loop *		
29		Generate all factors of a number.		
30		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. (Include selective import of modules and import * statements)		
31		Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.		
32		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.		

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
33		Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.		
34		Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.		
35		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.		
36		Write a Python program to read a file line by line and store it into a list.		
37		Write a Python program to read each row from a given csv file and print a list of string.		

PROGRAM 1

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

SOURCE CODE

```
startyear=2021
endyear=int(input('Enter the end year'))
print('The leap years are:')
for i in range(startyear,endyear):
    if(i%4==0 and i %100!=0 or i%400==0):
        print(i)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 leap.py
Enter the end year2030
The leap years are:
2024
2028
```

PROGRAM 2

AIM : List comprehensions:

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

(b) Square of N numbers.

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

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

SOURCE CODE

(a)

```
list=[2,3,4-5,0,7,8]
for num in list:
        if(num>0):
                print(num)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 3a.py
2
3
7
8
```

SOURCE CODE

(b)

```
numbers = [1, 2, 3, 4, 5]

s= [number ** 2 for number in numbers]

print(s)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 3b.py
[1, 4, 9, 16, 25]
```

SOURCE CODE

(c)

```
L=[]
s="India is my country"
for i in s:
  if i in ("aeiouAEIOU"):
    L.append(i)
print(L)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python name.py
['I', 'i', 'a', 'i', 'o', 'u']
```

SOURCE CODE

(d)

```
ordinal=input("Enter a name:")
print("The ASCII value of the letters in the word is")
for letter in ordinal:
n=ord(letter)
print(n)
```

OUTPUT

```
stud@debian:~/Anjana25/python$ python3 ord.py
112
108
97
121
105
110
103
```

PROGRAM 3

AIM : Count the occurence of each word in a line of text.

SOURCE CODE

```
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\HP\OneDrive\Desktop\python> python count.py
Enter a string:Sun rises in the east and sets in the west
Sun      1
rises    1
in       2
the      2
east     1
and      1
sets     1
west     1
```

PROGRAM 4

AIM : Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

SOURCE CODE

```
list=[]
while True:
        n=int(input("Enter the value:"))
        if(n<=100):
           list.append(n)
        else:
           list.append('over')
           print(list)
```

OUTPUT

```
stud@debian:~/Anjana25/python$ python3 liii.py
Enter the value:200
['over']
```

PROGRAM 5

AIM : Store a list of first names. Count the occurrence of ‘a’ within the list.

SOURCE CODE

```
l = ['a','j','aaa','aa']

print('occurance of a')

count = 0

for i in l:

        num=i.count('a')

        count=count+num

print(count)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn6.py
occurance of a
6
```

PROGRAM 6

AIM : Enter two list of integers. Check

(a) Whether they are of same length.

(b) Whether list sums to same value.

(c) Whether any value occur in both.

SOURCE CODE

```
l1=[5,6,3,7]
l2=[2,1,7,10,8]
x=len(l1)
y=len(l2)
if x==y:
  print("The list is of same length")
else:
  print("The list is of different length")
sum1=0
sum2=0
for i in range(len(l1)):
      sum1=sum1+l1[i]
print("The sum of list1 is:",sum1)
for j in range(len(l2)):
                  sum2=sum2+l2[j]
print("The sum of list2 is:",sum2)
if sum1=sum2:
      print("The sum of list1 is equal to list2")
else:
      print("The sum of list1 is not equal to list2")
```

```
for i in range(x):
        for j in range(y):
       if l1[i]==l2[j]:
          print(l1[i],"and",l2[j],"occur in both")
```

OUTPUT

```
stud@debian:~/Anjana25/python$ python3 occuu.py
The list is of different length
The sum of list1 is: 21
The sum of list2 is: 28
The sum of list1 is not equal to list2
7 and 7 occur in both
```

PROGRAM 7

AIM : Get a string from an input string where all occurence of first charecter replaced with '$', except first character.

[onion -> oni$n]

SOURCE CODE

```
ch=input("Enter a string:")
f=ch[0]
print(ch[0],end="")
f=f.lower()
for i in range(1,len(ch)):
 if ch[i]==f:
  print("$", end="")
 else:
  print(ch[i],end="")
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn8.py
Enter a string:onion
oni$nstud@debian:~/anjanpy$
```

PROGRAM 8

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

[eg : Python ->nythoP]

SOURCE CODE

```
s="python"
t=s[0]
t1=s[-1]
n=len(s)
ns=t1+s[1:n-1]+t
print(ns)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn9.py
nythop
```

PROGRAM 9

AIM: Accept the radius from user and find area of circle.

SOURCE CODE

```
p=int (input("Enter the radius"))
ar=3.14*p*p
print("Area=",ar)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn10.py
Enter the radius5
Area= 78.5
```

PROGRAM 10

AIM : Find biggest of 3 numbers entered .

SOURCE CODE

```
print("Enter 3 Numbers :")
a=int(input(""))
b=int(input(""))
c=int(input(""))
if (a>b)&(a>c):
        print(a,"is biggest")
if (b>a)&(b>c):
        print(b,"is biggest")
if (c>a)&(c>b):
        print(c,"is biggest")
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn11.py
enter first number 5
enter second number 12
enter third number 8
12 Is the greatest
```

PROGRAM 11

AIM: Accept a file name from user and print extension for that.

SOURCE CODE

```
import os
a=input("enter the filename : ")
print("The extension of file",a, "is",os.path.splitext(a))
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn12.py
enter the filename : python.py
The extension of file python.py is ('python', '.py')
```

PROGRAM 12

AIM: Create a list of colors from comma-separated colour names entered by user. Display first and last colours.

SOURCE CODE

```
n=int(input("Enter the size:"))
l=[]
for i in range(0,n):
        color=input("Enter Your Choice:")
        l.append(color)
print(l[0])
print(l[n-1])
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn13.py
Ente the size:4
Enter Your Choice:blue
Enter Your Choice:red
Enter Your Choice:green
Enter Your Choice:pink
blue
pink
```

PROGRAM 13

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

SOURCE CODE

```
num=input("Enter a number:")
dum1=num+num+num
dum2=num+num
dum3=num
print(int(dum1)+int(dum2)+int(dum3))
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn14.py
Enter a number:4
492
```

PROGRAM 14

AIM : Print out all colours from color list1 not contained in color list2.

SOURCE CODE

```
a=['red','green','blue','yellow']
b=['orange','pink','yellow','blue','violet']
for i in a:
        if i not in b:
                print(i)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 qn15.py
red
green
```

PROGRAM 15

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

SOURCE CODE

```
string1="Fisat"
string2="Ankamaly"
f1=string1[0]
f2=string2[0]
string=f2+string1[1:]+" "+f1+string2[1:]
print("The new string is :",string)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python swap.py
The new string is : Aisat Fnkamaly
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 16

AIM : Sort dictinary in ascending and descending order.

SOURCE CODE

```
dict1={"a":1,"c":3,"d":2,"b":4}

l=list(dict1.items())

print(l)

l.sort()

print("Ascending Order is \n",l)

l=list(dict1.items())

l.sort(reverse=True)

print("Descending order is \n",l) dict1={"a":1,"c":3,"d":2,"b":4}
l=list(dict1.items())
print(l)
l.sort()
print("Ascending order is\n",l)
l=list(dict1.items())
l.sort(reverse=True)
print("Descending order is\n",l)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python dictionary.py
[('a', 1), ('c', 3), ('d', 2), ('b', 4)]
Ascending order is
 [('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
 [('d', 2), ('c', 3), ('b', 4), ('a', 1)]
```

PROGRAM 17

AIM : Merge two dictionaries.

SOURCE CODE

```
D1={"name":"anju","age":"21"}
D2={"sex":"female","qualification":"bsc cs"}
D1.update(D2)
print(D1)
```

OUTPUT

```
stud@debian:~/Anjana25/python$ python3 profile.py
{'name': 'anjana', 'age': '20', 'sex': 'female', 'qualification': 'bca'}
```

PROGRAM 18

AIM : Find gcd of two numbers.

SOURCE CODE

```
x=int(input("Enter the first number:"))
y=int(input("Enter the second number:"))
if x>y:
   small=y
else:
   small=x
for i in range(1,small+1):
   if x%i==0 and y%i==0:
      hcf=i
print(hcf)
```

OUTPUT

```
stud@debian:~/anjanpy$ python3 gcd.py
Enter the first number:12
Enter the second number6
6
```

PROGRAM 19

AIM : Form a list of integers, create a list removing even numbers.

SOURCE CODE

```
list1=[]
list2=[]
n=int(input("Enter the list size :"))
for i in range(0,n):
        list1.append(int(input("Enter an element :")))
print("The list is\t",list1)
for i in list1:
        if i%2!=0:
                list2.append(i)
print("The odd list is\t",list2)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python list.py
Enter the list size :6
Enter an element :45
Enter an element :18
Enter an element :175
Enter an element :91
Enter an element :-62
Enter an element :425
The list is      [45, 18, 175, 91, -62, 425]
The odd list is  [45, 175, 91, 425]
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 20

AIM : Program to find the factorial of a number.

SOURCE CODE

```
n=int(input("Enter a Number :"))
factorial=1
for i in range(1,n+1):
        factorial=factorial*i
print("Factorial of",n,"=",factorial)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python list.py
Enter a Number :6
Factorial of 6 = 720
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 21

AIM : Generate Fibonacci series of N terms.

SOURCE CODE

```
n=int(input("Enter a Number :"))
print("The first",n,"fibonacci seriers is :")
f1=0
f2=1
for i in range(0,n):
        print(f1)
        f3=f1
        f1=f1+f2
        f2=f3
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python fib.py
Enter a Number :6
The first 6 fibonacci seriers is :
0
1
1
2
3
5
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 22

AIM : Find the sum of all items in a list.

SOURCE CODE

```
list1=[1,2,3,4,5,6,7]
summ=0
for i in list1:
        summ=summ+i
print("sum=",summ)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python sum.py
sum= 28
PS C:\Users\HP\OneDrive\Desktop\python>
```

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.

SOURCE CODE

```
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(list1)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python digit.py
[4624, 6084, 6400, 8464]
PS C:\Users\HP\OneDrive\Desktop\python> []
```

PROGRAM 24

AIM : Display the given pyramid with step numbers accepted from user.

Eg : 4

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

SOURCE CODE

```
n=int(input("Enter a number :"))
for i in range(1,n+1):
        for j in range(i,(i*i)+1,i):
                print(j,"\t",end="")
        print("\n")
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python pt.py
Enter a number :4
1

2       4

3       6       9

4       8       12      16
```

PROGRAM 25

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

SOURCE CODE

```
string=input("Enter a string :")
ulist=[]
for i in string:
    if i not in ulist:
        ulist.append(i)
for i in ulist:
    count=0
    for j in string:
        if(i==j):
            count+=1
    print(i,"\t:",count)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python count.py
Enter a string :anjana
a       : 3
n       : 2
j       : 1
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 26

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

SOURCE CODE

```
string=input("Enter a string :")
if(string[-3:]=="ing"):
        string+="ly"
else:
        string+="ing"
print(string)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python str.py
Enter a string :I love India
I love Indiaing
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 27

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

SOURCE CODE

```
wlist=[]
print("Enter 5 words :")
for i in range(0,5):
        wlist.append(input(""))
temp=wlist[0]
for i in range(1,5):
        if len(wlist[i])>len(temp):
                temp=wlist[i]
print("Length of longest word is",len(temp))
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python list.py
Enter 5 words :
good
morning
sun
stars
cloud
Length of longest word is 7
```

PROGRAM 28

AIM : Construct following pattern using nested loop.

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

SOURCE CODE

```
for i in range(1,6):
    for j in range(0,i):
        print("*",end="")
    print("\n")
for i in range(4,0,-1):
    for j in range(0,i):
        print("*",end="")
    print("\n")
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python pt.py
*
**
***
****
*****
****
***
**
*
```

PROGRAM 29

AIM : Generate all factors of a number.

SOURCE CODE

```
n=int(input("Enter a number :"))
print("The factors are :")
for i in range(1,n+1):
        if(n%i)==0:
                print(i)
```

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python> python fac.py
Enter a number :6
The factors are :
1
2
3
6
PS C:\Users\HP\OneDrive\Desktop\python>
```

PROGRAM 30

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

Terminal Commands

```
PS D:\mySpace\learn> cd python
PS D:\mySpace\learn\python> md Graphics

    Directory: D:\mySpace\learn\python


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        28-02-2022  08.29 PM                Graphics


PS D:\mySpace\learn\python> cd Graphics
PS D:\mySpace\learn\python\Graphics> notepad __init__.py
PS D:\mySpace\learn\python\Graphics> notepad circle.py
PS D:\mySpace\learn\python\Graphics> notepad rectangle.py
PS D:\mySpace\learn\python\Graphics> md tdgraphics


    Directory: D:\mySpace\learn\python\Graphics


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        28-02-2022  08.32 PM                tdgraphics


PS D:\mySpace\learn\python\Graphics> cd tdgraphics
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad __init__.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad cuboid.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad sphere.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> cd ..
PS D:\mySpace\learn\python\Graphics> cd ..
PS D:\mySpace\learn\python>
```

SOURCE CODE

Graphice\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 10 is : ",circle.area_circle(10))
print("Permeter of a circle with radius 10 is ",circle.perimeter_circle(10))
print("\n")


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


print("Area of a  cuboid with length,width,height 10 is : ",cuboid.area_cuboid(10,10,10))
print("Volume of a cuboid with length,width,height 10 is : ",cuboid.volume_cuboid(10,10,10))
print("\n")


print("Area of a spere with radius 10 is : ",sphere.area_sphere(10))
print("Permeter of a spere with radius 10 is ",sphere.perimeter_sphere(10))
```

OUTPUT

```
PS D:\mySpace\learn\python> python graphics.py
Area of a circle with radius 10 is :  314.1592653589793
Permeter of a circle with radius 10 is  62.83185307179586


Area of a Rectangle with length and width 10 is :  100
Permeter of a Rectangle with length and width 10 is :  40


Area of a cuboid with length,width,height 10 is :  600
Volume of a cuboid with length,width,height 10 is :  1000


Area of a spere with radius 10 is :  1256.6370614359173
Permeter of a spere with radius 10 is  62.83185307179586
PS D:\mySpace\learn\python>
```

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.

SOURCE CODE

```
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))
        def print(self):
                print(self.area)

r1=Rectangle(10,2)
r2=Rectangle(5,8)
x=r1.area()
y=r2.area()
print("area of first rectangle is",x)
print("area of second rectangle is",y)
p=r1.perimeter()
q=r2.perimeter()
print("perimeter of first rectangle is",p)
print("perimeter of second rectangle is",q)
print
if(x>y):
        print('Area of first rectangle greater than second rectangle');
else:
        print('Area of second rectangle greater than first rectangle');
```

OUTPUT

```
area of first rectangle is 20
area of second rectangle is 40
perimeter of first rectangle is 24
perimeter of second rectangle is 26
Area of second rectangle greater than first rectangle
```

PROGRAM 32

AIM: Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposite at the bank and withdraw an amount from the bank.

SOURCE CODE

```
class Bank:
        def __init__(self,accno,aname,a_type,bal):
                self.accno=accno
                self.aname=aname
                self.a_type=a_type
                self.bal=bal
        def withdraw(self,x):
                self.bal=self.bal-x
        def deposit(self,y):
                self.bal=self.bal+y
        def print(self):
                print(self.accno,self.aname,self.a_type,self.bal)
acc1=Bank(2435,'anju','sbi',10000)
acc2=Bank(5436,'aju','federal',22000)
acc1.withdraw(1000)
acc1.deposit(4000)
acc2.withdraw(2500)
acc1.deposit(8000)
acc1.print()
acc2.print()
```

OUTPUT

```
2435 anju sbi 21000
5436 aju federal 19500
```

PROGRAM 33

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

SOURCE CODE

```
class Rectangle:
        def __init__(self,ln,br):
                self.ln=ln
                self.br=br
        def area(self):
                p=self.ln*self.br
                return p
        def __lt__(self,r1):
                if r2.area()<r1.area():
                        return r2.area()
                else:
                        return r1.area()

        def perim(self):
                q=2*(self.ln+self.br)
                return q

a=int(input("Enter length of the first rectangle:"))
b=int(input("Enter breadth of the first rectangle:"))
r1=Rectangle(a,b)
a=int(input("Enter length of the second rectangle:"))
b=int(input("Enter breadth of the second rectangle:"))
r2=Rectangle(a,b)

print("Perimeter of first rectangle= ",r1.perim())
print("Perimeter of second rectangle= ",r2.perim())

print("Least one is:",r1<r2)
```

OUTPUT

```
Enter length of the first rectangle:12
Enter breadth of the first rectangle:4
Enter length of the second rectangle:8
Enter breadth of the second rectangle:6
Perimeter of first rectangle=  32
Perimeter of second rectangle=  28
Least one is: 48
```

PROGRAM 34

AIM : Create a class Time with private attributes hour,minute and second. Overload '+' operator to find sum of two time.

SOURCE CODE

```
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(hr,":",min,":",sec)

t1=Time(3,45,56)
t2=Time(4,20,3)
t1+t2
```

OUTPUT

```
7 : 65 : 59
```

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

SOURCE CODE

```
class Publisher:
    def __init__(self,name):
        self.name=name
class Book(Publisher):
    def __init__(self,name,title,auther):
        super().__init__(name)
        self.title=title
        self.auther=auther
    def print_function(self):
        print("This Fuction is a member fuction of class Publisher")
class Python(Book):
    def __init__(self,name,title,auther,price,nop):
        super().__init__(name,title,auther)
        self.price=price
        self.nop=nop
    def print_function(self):
        print("Name :",self.name)
        print("Title :",self.title)
        print("Auther :",self.auther)
        print("Price :",self.price)
        print("Number of Pages :",self.nop)
```

```
p1=Python("Text book","Python Programming","Mr.abc",100,500)
p1.print_function()
p2=Book("a","b","c")
p2.print_function()
```

OUTPUT

```
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 100
Number of Pages : 500
This Fuction is a member fuction of class Publisher
```

PROGRAM 36

AIM: Write a program to read a file line by line and store it into a list

SOURCE CODE

```
fp=open("text_file.txt",'r')
lines=[]
for line in fp:
    lines.append(line.strip())
print(lines)
```

text.txt

"Cats, also called domestic cats are small, carnivorous mammals, of the family Felidae.

Domestic cats are often called 'house cats' when kept as indoor pets.

Cats have been domesticated for nearly 10,000 years.

They are one of the most popular pets in the world."

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn1.py
['"Cats, also called domestic cats are small, carnivorous mammals, of the family Felidae.', "Domestic cat
s are often called 'house cats' when kept as indoor pets.", 'Cats have been domesticated for nearly 10,00
0 years.', 'They are one of the most popular pets in the world."']
PS C:\Users\HP\OneDrive\Desktop\python\co5>
```

PROGRAM 37

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

SOURCE CODE

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

text.csv

Name,Designation,Salary

Jessy,Manager,90000

Tom,Clerk,40000

Alfred,Assistant Manager,70000

OUTPUT

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn2.py
['Name', 'Designation', 'Salary']
['Jessy', 'Manager', '90000']
['Tom', 'Clerk', '40000']
['Alfred', 'Assistant Manager', '70000']
PS C:\Users\HP\OneDrive\Desktop\python\co5>
```