FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB

LABORATORY RECORD

Name: ANN MARIYA T M

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1   Batch: A   Roll No: 28

University Registration Number: FIT21MCA-2029

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

<u>CERTIFICATE</u>

This is to certify that this is a Bonafide record of the Practical work done by **ANN MARIYA T M (FIT21MCA-2029)** *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   Signature of H O D

Name: Name:

Date of University practical examination ………………………

Signature of   Signature of
Internal Examiner   External Examiner

CONTENT

SI No:	Date :	Name of Experiment:	Page No:	Signature of Staff –In – Charge:
		COURSE OUTCOME 1		
1	28/10/2021	Display future leap years from current year to a final year entered by user.	5	
2	28/10/2021	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)	5	
3	28/10/2021	Count the occurrences of each word in a line of text.	7	
4	28/10/2021	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	8	
5	10/11/2021	Store a list of first names. Count the occurrences of 'a' within the list	9	
6	10/11/2021	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.	9	
7	10/11/2021	Get a string from an input string where all occurrences of first character replaced with '$', except first character	11	
8	10/11/2021	Create a string from given string where first and last characters exchanged. [eg: python - > nythop]	11	
9	10/11/2021	Accept the radius from user and find area of circle.	12	
10	11/11/2021	Find biggest of 3 numbers entered.	12	
11	11/11/2021	Accept a file name from user and print extension of that.	12	

12	11/11/2021	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	13	
13	11/11/2021	Accept an integer n and compute n+nn+nnn.	13	
14	11/11/2021	Print out all colors from color-list1 not contained in color-list2.	14	
15	17/11/2021	Create a single string separated with space from two strings by swapping the character at position 1.	14	
16	17/11/2021	Sort dictionary in ascending and descending order.	15	
17	17/11/2021	Merge two dictionaries.	15	
18	17/11/2021	Find gcd of 2 numbers.	16	
19	17/11/2021	From a list of integers, create a list removing even numbers.	16	

COURSE OUTCOME 217

20	25/11/2021	Program to find the factorial of a number	17	
21	25/11/2021	Generate Fibonacci series of N terms	17	
22	25/11/2021	Find the sum of all items in a list	18	
23	25/11/2021	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	18	
24	2/12/2021	Display the given pyramid with step number accepted from user.	19	
25	2/12/2021	Count the number of characters (character frequency) in a string.	20	
26	2/12/2021	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.	21	

27	9/12/2021	Accept a list of words and return length of longest word.	21	
28	9/12/2021	Construct following pattern using nested loop. * * * * * * * * * * * * * * * * * * * * * * * * *	22	
29	9/12/2021	Generate all factors of a number.	23	

COURSE OUTCOME 3

30	29/01/2022	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)	24	

COURSE OUTCOME 4

31	13/01/2022	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	27	
32	13/01/2022	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.	28	

33	13/01/2022	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	30	
34	20/01/2022	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	31	
35	20/01/2022	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.	32	

COURSE OUTCOME 534

36	03/02/2022	Write a Python program to read a file line by line and store it into a list.	34	
37	03/02/2022	Write a Python program to read each row from a given csv file and print a list of strings.	34	

COURSE OUTCOME 1

1) **Display future leap years from current year to a final year entered by User.**

   Source code

   ```
   print("print leap year
   between two given years");
   startyear=2021
   endyear=int(input("Enter end year")) print("list of leap years")
   for year in
       range(startyear,endyear
       ): if(0==year%4):
           print(year)
   ```

   Output

   ```
   C:\Users\Ann mariya T M\Desktop\Python>python leap.py
   print leap year between two given years
   Enter end year2035
   list of leap years
   2024
   2028
   2032
   ```

2) **List comprehensions:**

   a. **Generate positive list of numbers from a given list of integers.**

   Source code

   ```
   list=[-11,4,8,-34,10,14]
   print("Elements in the list are:",list) print("Positive numbers in the list")
   for num in list:
       if num>=0:
           print(num)
   ```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python list_positive.py
Elements in the list are:  [-11, 4, 8, -34, 10, -14]
Positive numbers in the list
4
8
10
```

b. Square of N numbers

Source code

n=int(input('Enter range:'))

for num in range(1,n+1):

    num=num*num

    print(num)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python list_square.py
Enter range:6
1
4
9
16
25
36
```

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

Source code

s=input("Enter a string: ")

list=[]

for i in s:

   if i in "aeiouAEIOU":

        list.append(i)

print("vowels in the list are:")

print(list)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python list_vowel.py
Enter a string: hello
vowels in the list are:
['e', 'o']
```

    d. **List ordinal values of each element of a word.**

    Source code

```
print("String: Welcome")
print("Ordinal Values")
for i in 'W','e','l','c','o','m','e':
    x=ord(i)
    print(x)
```

    Output

```
C:\Users\Ann mariya T M\Desktop\Python>python ordinal.py
String: Welcome
Ordinal Values
87
101
108
99
111
109
101
```

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

    Source code

```
list1=[]
list2=[]
x=input("Enter a line of text:")
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

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_4.py
Enter a line of text:Do good and good will come to you.
Do        1
good      2
and       1
will      1
come      1
to        1
you.      1
```

4) **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 an integer: '))

    if(n<=100):

          list.append(n)

    else:

          list.append('over')

          print(list)

   Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_4.py
Enter an integer: 23
Enter an integer: 36
Enter an integer: 104
[23, 36, 'over']
Enter an integer: 34
Enter an integer: 53
Enter an integer: 200
[23, 36, 'over', 34, 53, 'over']
```

5) Store a list of first names. Count the occurrences of 'a' within the list.

Source code

```
list=['ann','mariya','anju'] print("Elements in the list are:")
print(list)
count=0
for word in list:
        for i in word:
                if i=='a':
                        count+=1
print("count of 'a' is:", count)
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python 6occurence.py
Elements in the list are:
['ann', 'mariya', 'anju']
count of 'a' is: 4
```

6) Enter 2 lists of integers.Check

    a. whether list are of same length

    b. whetherlist sums of same value

    c. whether any value occur in both.

Source code

```
l1=[1,2,3,4]
l2=[1,3,2]
print("List 1",l1)
print("List 2",l2)
x=len(l1)
y=len(l2)
if x==y:
 print("List are of same length")
else:
```

```python
   print("Length of lists are different")

s1=0

s2=0

for i in range(x):
 s1=s1+l1[i]

print("Sum of elements of List1:",s1)

for j in range(y):
 s2=s2+l2[j]

print("Sum of elememts of List2:",s2)

if s1==s2:
 print("Sum of list elements is same")

else:
 print("Sum of list elements is not same")

print("Common elements are:")

for i in range(x):
 for j in range(y):
        if l1[i]==l2[j]:

                print(l1[i])
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python 7list.py
List 1 [1, 2, 3, 4]
List 2 [1, 3, 2]
Length of lists are different
Sum of elements of List1: 10
Sum of elememts of List2: 6
Sum of list elements is not same
Common elements are:
1
2
3
```

7) Get a string from an input string where all occurrences of first character replaced with '$',except first character.[eg:onion->oni$n]

Source code

str=input("Enter a string: ")

print("Original string is: ",str)

char=str[0]

str=str.replace(char,'$')

str=char+str[1:]

print("String: ",str)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_8.py
Enter a string: occupation
Original string is:  occupation
String:  occupati$n
```

8) Create a string from given string where first and last characters exchanged.
[eg:python->nythop]

Source code

s=input("Enter a string: ")

t=s[0]

t1=s[-1]

n=len(s)

ns=t1+s[1:n-1]+t

print(ns)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_9.py
Enter a string: python
nythop

C:\Users\Ann mariya T M\Desktop\Python>python CO1_9.py
Enter a string: hello
oellh
```

9) Accept the radius from the user and find the area of the circle.

Source code

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

A=3.14*r*r

print(A)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python radius_circle.py
Enter the radius: 6
113.03999999999999
```

10) Find the biggest of 3 numbers

Source code

a=int(input('Enter first number:'))

b=int(input('Enter second number:'))

c=int(input('Enter third number:'))

if a>b and a>c:

 print(a)

if b>a and b>c:

 print(b)

if c>a and c>b:

 print(c)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_11.py
Enter first number:4
Enter second number:6
Enter third number:2
6
```

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

Source code

import os

a=input("Enter file name:")

print("The extension of file",a,"is",os.path.splitext(a))

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python file_extension.py
Enter file name:Exam.xls
The extension of file Exam.xls is ('Exam', '.xls')
```

12) Create a list of colors from comma-separated color names entered by user.

Display first and last colors.

Source code

colors=[]

str=(input("Enter color names:"))

for i in str.split(','):

 colors.append(i)

print(colors)

print("first color:",colors[0],"Last color:",colors[-1])

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_13.py
Enter color names:yellow,red,blue,pink
['yellow', 'red', 'blue', 'pink']
first color: yellow Last color: pink
```

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

Source code

n=int(input("Enter the number:"))

a=n*1

b=n*11

c=n*111

s=a+b+c

print(n,"+",n,"*",n,"+",n,"*",n,"*",n,"=",s)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_14.py
Enter the number:5
5 + 5 * 5 + 5 * 5 * 5 = 615
```

14) Print out all color from color-list1 not contained in color-list2

Source code

l1=['red','green','blue','yellow','black']

l2=['red','green','yellow']

print(l1)

print(l2)

print("Colors that are not in l1:

")

for i in l1:

 if i not in l2:

        print(i)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_15.py
['red', 'green', 'blue', 'yellow', 'black']
['red', 'green', 'yellow']
Colors that are not in l1:
blue
black
```

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

Source code

str1=input("Enter first string:")

str2=input("Enter second string:")

str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]

print(str3)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_16.py
Enter first string:Ann
Enter second string:Mariya
Mnn Aariya
```

16) Sort dictionary in ascending and descending order.

Source code

d1={"a":1,"c":3,"d":2,"b":4}

l=list(d1.items())

print(l)

l.sort()

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

l=list(d1.items())

l.sort(reverse=True)

print("Descending order is\n",l)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_17.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)]
```

17) Merge two dictionaries.

Source code

D1={"Name":"Ann mariya","Age":"20"}

print("Directory 1",D1)

D2={"Gender":"Female","Qualification":"BCA"}

print("Directory 2",D2)

D1.update(D2)

print("After merging...")

print(D1)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_18.py
Directory 1 {'Name': 'Ann mariya', 'Age': '20'}
Directory 2 {'Gender': 'Female', 'Qualification': 'BCA'}
After merging...
{'Name': 'Ann mariya', 'Age': '20', 'Gender': 'Female', 'Qualification': 'BCA'}
```

18) Find gcd of 2 numbers

Source code

```
a=int(input("Enter first number: "))
b=int(input("Enter first number: "))
x=min(a,b)
gcd=0
for i in range (1,x+1):
 if((a%x==0) and (b%x==0)):
        gcd=i
print("GCD is",i)
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_19.py
Enter first number: 6
Enter first number: 8
GCD is 6
```

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

Source code
```
l1=[1,2,3,4,5,6,7,8,9,10]
print(l1)
l2=[]
for i in range(len(l1)):
 if l1[i]%2!=0:
        l2.append(l1[i])
print("List after removing even elements")
print(l2)
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO1_20.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
List after removing even elements
[1, 3, 5, 7, 9]
```

COURSE OUTCOME 2

20) **Program to find the factorial of a number.**

Source code

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

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python factorial.py
Enter a number:6
720

C:\Users\Ann mariya T M\Desktop\Python>python factorial.py
Enter a number:4
24
```

21) **Generate fibonacci series of N terms.**

Source code

```
n=int(input('Enter a limit:'))
a=0
b=1
print(a)
print(b)
for i in range (2,n):
        c=a+b
        print(c)
        a=b
        b=c
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python fibo.py
Enter a limit:6
0
1
1
2
3
5
```

22) **Find the sum of all items in a list.**

Source code

list=[2,6,9,11,25]

print("List elements are:",list)

sum=0

for i in list:

       sum=sum+i

print("The sum of list elements is:",sum)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_3.py
List elements are: [2, 6, 9, 11, 25]
The sum of list elements is: 53
```

23) **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(k)
print(list1)
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_4.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

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

Source code

```
n=int(input("Enter a number:"))
for j in range(0,n+1):
    for i in range(1,j+1):
            i=j*i
            print(i,end=" ")
    print("\n")
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_5.py
Enter a number:4


1

2 4

3 6 9

4 8 12 16
```

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

Source code

```
string=input("Enter a string:")

list1=[]

for i in string:

    if i not in list1:

            list1.append(i)

for i in list1:

    count=0

    for j in string:

            if(i==j):

                    count=count+1

    print(i,"\t:",count)
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_6.py
Enter a string:welcome
w        : 1
e        : 2
l        : 1
c        : 1
o        : 1
m        : 1
```

26) 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

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_7.py
Enter a string:Dancing
Dancingly

C:\Users\Ann mariya T M\Desktop\Python>python CO2_7.py
Enter a string:Dance
Danceing
```

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

Source code

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

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_8.py
Enter the range:3
Enter the words:
Hello
Hai
Good
Length of longest word is 5
```

28) Construct following pattern using nested loop.

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

Source code

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

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_9.py
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

29) **Generate all factors of a number.**

Source code

n=int(input("Enter a number:"))

print("Factors are")

for i in range(1,n+1):

   if(n%i==0):

        print(i)

Output

```
C:\Users\Ann mariya T M\Desktop\Python>python CO2_10.py
Enter a number:5
Factors are
1
5

C:\Users\Ann mariya T M\Desktop\Python>python CO2_10.py
Enter a number:8
Factors are
1
2
4
8
```

COURSE OUTCOME 3

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)**

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

```
C:\Users\Ann mariya T M\Desktop>cd Python

C:\Users\Ann mariya T M\Desktop\Python>md Graphics

C:\Users\Ann mariya T M\Desktop\Python>cd Graphics

C:\Users\Ann mariya T M\Desktop\Python\Graphics>notepad __init__.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics>notepad circle.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics>notepad rectangle.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics>md tdgraphics

C:\Users\Ann mariya T M\Desktop\Python\Graphics>cd tdgraphics

C:\Users\Ann mariya T M\Desktop\Python\Graphics\tdgraphics>notepad __init__.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics\tdgraphics>notepad cuboid.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics\tdgraphics>notepad sphere.py

C:\Users\Ann mariya T M\Desktop\Python\Graphics\tdgraphics>cd ..

C:\Users\Ann mariya T M\Desktop\Python\Graphics>cd ..
```

```
C:\Users\Ann mariya T M\Desktop\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
```

COURSE OUTCOME 4

31) **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,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)

l=int(input("Enter length of rectangle1: "))

b=int(input("Enter breadth of rectangle1: "))

rect1 = Rectangle(l,b)

a1=rect1.area()

p1=rect1.perimeter()

print("Area:",a1)

print("Perimeter:",p1)

l=int(input("Enter length of rectangle2: "))

b=int(input("Enter breadth of rectangle2: "))

rect2 = Rectangle(l,b)

a2=rect2.area()

p2=rect2.perimeter()

print("Area:",a2)
```

```
print("Perimeter:",p2)

if (a1>a2):

    print("First rectangle is larger")

elif a1==a2:

    print("Rectangles are of same area")

else:

    print("Second rectangle is larger")
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_1.py
Enter length of rectangle1: 4
Enter breadth of rectangle1: 6
Area: 24
Perimeter: 20
Enter length of rectangle2: 2
Enter breadth of rectangle2: 3
Area: 6
Perimeter: 10
First rectangle is larger
```

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.**

Source code

```
class bank:
def __init__(self,acc_no,name,acc_type,bal):

        self.acc_no=acc_no

        self.name=name

        self.acc_type=acc_type

        self.bal=bal

    def deposit(self):
```

```python
        self.bal=self.bal+y

        return self.bal

    def withdraw(self):

        return self.bal-y

    def display_balance(self):

        return self.bal

acc1=bank("b11","Ann","Savings",50000)

while(1):

    print("1.Deposit\n2.Withdraw\n3.Display balance\n4.Exit\n")

    ch=int(input("Enter your choice:"))

    if ch==1:

        amt=int(input("Enter the amount:"))

        b=acc1.deposit(amt)

        print("Current balance:",b)

    elif ch==2:

        amt=int(input("Enter the amount:"))

        b=acc1.withdraw(amt)

        print("Current balance:",b)

    elif ch==3:

        cb=acc1.display_balance()

        print("Current balance:",cb)

    elif ch==4:

        exit(1)

    else:
```

```
        print("Invalid choice")
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_2.py
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:3
Current balance: 50000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:1
Enter the amount:2000
Current balance: 52000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:2
Enter the amount:1000
Current balance: 51000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:4

C:\Users\Ann mariya T M\Desktop\Python\CO4>
```

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

Source code

class Rectangle:

    def __init__(self,length,breadth):

        self.__length = length

        self.__breadth = breadth

    def __lt__ (self,rect2):

        if self.__length*self.__breadth < rect2.__length*rect2.__breadth:

            return True

        else:

return False

l=int(input("Enter length of rectangle1: "))

b=int(input("Enter breadth of rectangle1: "))

rect1 = Rectangle(l,b)

l=int(input("Enter length of rectangle2: "))

b=int(input("Enter breadth of rectangle2: "))

rect2 = Rectangle(l,b)

if rect1 < rect2:

    print("Second rectangle is larger")

else:

    print("First rectangle is larger")

output

```
C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_3.py
Enter length of rectangle1: 1
Enter breadth of rectangle1: 3
Enter length of rectangle2: 5
Enter breadth of rectangle2: 8
Second rectangle is larger

C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_3.py
Enter length of rectangle1: 6
Enter breadth of rectangle1: 9
Enter length of rectangle2: 2
Enter breadth of rectangle2: 4
First rectangle is larger
```

34) Create a class Time with private attributes hour, minute and second.
Overload '+' operator to find sum of 2 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(3,35,56)

   t2=Time(4,20,3)

   print('Time 1: 3:35:56')

   print('Time 2: 4:20:3')

   print('Adding........')

   t1+t2
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_4.py
Time 1: 3:35:56
Time 2: 4:20:3
Adding.......
7 : 55 : 59
```

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.**
   Source code
   class Publisher(object):
       def __init__(self,name):
           self.name=name
       def display1(self):

```python
        print(self.title)
        print(self.author)
class Book(Publisher):
    def __init__(self,name,title,author):
        super().__init__(name)
        self.title=title
        self.author=author
    def display2(self):
        #super().display1()
        print(self.title)
        print(self.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 display3(self):
        super().display2()
        print(self.price)
        print(self.no_of_pages)
p=Python("ABC Publications","Taming Python","jeeva jose",100,500)
p.display3()
q=Python("XYZ Publications","Java programming","E
Balagurusami",500,1200)
q.display3()
```

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO4>python CO4_5.py
Taming Python
jeeva jose
100
500
Java programming
E Balagurusami
500
1200
```

COURSE OUTCOME 5

36) Write a Python 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)

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO5>python 1.py
["Kerala, a state on India's tropical Malabar Coast, has nearly 600km of Arabian
 Sea shoreline. It's known for its palm-lined beaches and backwaters, a network
of canals. Inland are the Western Ghats, mountains whose slopes support tea, cof
fee and spice plantations as well as wildlife."]
```

37) **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)

Output

```
C:\Users\Ann mariya T M\Desktop\Python\CO5>python 2.py
['Name', 'Age', 'Profession']
['John', '30', 'Manager']
['Jerin', '20', 'Accountant']
['Ann', '22', 'Professor']
['Angel', '24', 'Engineer']
['Sree lakshmi', '28', 'Doctor']
```