FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

LABORATORY RECORD

20MCA131 - PROGRAMMING LAB

Name: **ASHNA SHERIN A M**

Branch: **MASTER OF COMPUTER APPLICATION**

Semester: **1** **Batch: 2021 A** **Roll No: 40**

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name : ASHNA SHERIN A M

Branch : MASTER OF COMPUTER APPLICATIONS

Semester : 1 **Roll No: 40**

University Exam.Reg. No: FIT21MCA-2040

CERTIFICATE

Certified that this is the Bonafede record of the Practical work done by Mr. **ASHNA SHERIN A M (FITMCA2040)** in the **20MCA131-PROGRAMMING** Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.

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

Name: — Name:

Date:

Date of University practical examination ……

Signature of External Examiner — Signature of Internal 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.	**7**	
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)	**7**	
3	**28/10/2021**	Count the occurrences of each word in a line of text.	**9**	
4	**28/10/2021**	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	**10**	
5	**10/11/2021**	Store a list of first names. Count the occurrences of 'a' within the list	**10**	
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.	**11**	
7	**10/11/2021**	Get a string from an input string where all occurrences of first character replaced with '$', except first character	**12**	
8	**10/11/2021**	Create a string from given string where first and last characters exchanged. [eg: python - > nythop]	**13**	
9	**10/11/2021**	Accept the radius from user and find area of circle.	**13**	
10	**11/11/2021**	Find biggest of 3 numbers entered.	**13**	

11	**11/11/2021**	Accept a file name from user and print extension of that.	**14**	
12	**11/11/2021**	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	**14**	
13	**11/11/2021**	Accept an integer n and compute n+nn+nnn.	**15**	
14	**11/11/2021**	Print out all colors from color-list1 not contained in color-list2.	**15**	
15	**17/11/2021**	Create a single string separated with space from two strings by swapping the character at position 1.	**16**	
16	**17/11/2021**	Merge two dictionaries.	**16**	
17	**17/11/2021**	Find gcd of 2 numbers.	**17**	
18	**17/11/2021**	From a list of integers, create a list removing even numbers.	**17**	
COURSE OUTCOME-2				
19	**25/11/2021**	Program to find the factorial of a number	**18**	
20	**25/11/2021**	Generate Fibonacci series of N terms	**19**	
21	**25/11/2021**	Find the sum of all items in a list	**19**	
22	**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.	**20**	
23	**2/12/2021**	Display the given pyramid with step number accepted from user.	**21**	
24	**2/12/2021**	Count the number of characters (character frequency) in a string.	**22**	
25	**2/12/2021**	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.	**22**	

26	**09/12/2021**	Accept a list of words and return length of longest word.	**23**	
27	**09/12/2021**	Construct following pattern using nested loop. *	**24**	
28	**09/12/2021**	Generate all factors of a number.	**25**	
COURSE OUTCOME-3				
29	**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)	**26**	
COURSE OUTCOME-4				
30	**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.	**29**	
31	**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.	**30**	

32	**13/01/2022**	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	**31**	
33	**20/01/2022**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	**32**	
34	**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.	**33**	
		COURSE OUTCOME-5		
35	**03/02/2022**	Write a Python program to read a file line by line and store it into a list.	**34**	
36	**03/02/2022**	Write a Python program to read each row from a given csv file and print a list of strings.	**35**	

COURSE OUTCOME 1

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

Source code

```
print("Enter leap year between given two 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

```
print leap year between two given years
Enter startyear2000
Enter end year2020
list of leap years
2000
2004
2008
2012
2016
```

2) List comprehensions:

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

Source code

```
list=[22,-56,8,-5,7,14]
for num in list:
  if num>=0:
    print(num)
```

Output

```
stud@debian:~/python$ python3 list1.py
0
3
4
5
```

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

```
stud@debian:~/python$ python3 list2.py
[1, 4, 9, 16, 25]
```

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

```
stud@debian:~/python$ python3 list3.py
['a']
['a', 'e']
```

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

Source code

```
print("String: Hallo")
print("Ordinal Values")
for i in 'H','a','l','l','o':
   x=ord(i)
   print(x)
```

Output

```
stud@debian:~/python$ python3 list4.py
Enter the word :ashna
[97, 115, 104, 110, 97]
```

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

```
stud@debian:~/python$ python3 3.py
Enter a line of text:hiii hello
hiii     1
hello    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

```
stud@debian:~/python$ python3 4.py
Enter an integer: 12
Enter an integer: 15
Enter an integer: 100
Enter an integer: 105
[12, 15, 100, 'over']
```

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

Source code

```
list=['anu','sherin','ashna']
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

```
stud@debian:~/python$ python3 5.py
Elements in the list are:
['anu', 'sherin', 'ashna']
count of 'a' is: 3
```

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=[3,7,9,7]
 l2=[6,2,2,2]
 print(l1)
 print(l2)
 flag=0
#To check whether list are of same length
if len(l1)==len(l2):
        print('list are of same length\n')
else:
        print('list length is different\n')
#whether list sums to same value
 sum1=0
 sum2=0
 for i in l1:
      sum1=sum1+i
print("The sum of list one is ",sum1)
for j in range (len(l2)):
        sum2=sum2+l2[j]
print("The sum of list two is ",sum2);
```

```
if sum1==sum2:
        print("the sum of 2 lists are same\n")
else:
        print("the sum of 2 lists are not same\n")
#whether any value occurs in both
for i in l1:
        if i in l2:
                print("occurs in both list ",i)
                flag=1
if flag==0:
        print("no common elements")
```

Output

```
[3, 7, 9, 7]
[6, 2, 2, 2]
list are of same length

The sum of list one is  26
The sum of list two is  12
the sum of 2 lists are not same

no common elements
```

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

```
stud@debian:~/python$ python3 7.py
Enter a string: onion
Original string is:  onion
String:  oni$n
stud@debian:~/python$
```

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

```
stud@debian:~/python$ python3 8.py
Enter a string: python
nythop
```

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

```
stud@debian:~/python$ python3 9.py
Enter the radius: 7
153.86
```

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

```
stud@debian:~/python$ python3 lar.py
Enter first number:6
Enter second number:9
Enter third number:23
23
```

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

```
stud@debian:~/python$ python3 11.py
Enter file name:8.py
The extension of file 8.py is ('8', '.py')
stud@debian:~/python$ python3 11.py
```

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

```
stud@debian:~/python$ python3 12.py
Enter color names:white black blue
['white black blue']
first color: white black blue Last color: white black blue
```

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

```
stud@debian:~/python$ python3 13.py
Enter the number:15
15 + 15 * 15 + 15 * 15 * 15 = 1845
```

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

```
stud@debian:~/python$ python3 14.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

```
stud@debian:~/python$ python3 15.py
Enter first string:ashna
Enter second string:sherin
sshna aherin
```

16) 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

```
Directory 1 {'Name': 'Ashna', 'Age': '21'}
Directory 2 {'Gender': 'Female', 'Qualification': 'BCA'}
After merging...
{'Name': 'Ashna', 'Age': '21', 'Gender': 'Female', 'Qualification': 'BCA'}
```

17) Find gcd of 2 numbers

Source code

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

Output

```
stud@debian:~/python$ python3 17.py
Enter a value : 34
Enter second value: 33
gcd is  1
```

18) 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

```
stud@debian:~/python$ python3 18.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
List after removing even elements
[1, 3, 5, 7, 9]
```

COURSE OUTCOME 2

19) 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

```
stud@debian:~$ python3 new.py
enter the value:5
120
stud@debian:~$
```

20) 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

```
stud@debian:~$ python3 new.py
enter the value:7
0
1
1
2
3
5
8
stud@debian:~$
```

21) Find the sum of all items in a list.

Source code

```
list=[2,8,9,34,25]
print("List elements are:",list)
sum=0
for i in list:
        sum=sum+i
print("The sum of list elements is:",sum)
```

Output

```
stud@debian:~$ python3 new.py
List elements are: [5, 6, 3, 4, 7]
The sum of list elements is: 25
stud@debian:~$
```

22) 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

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

23) 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

```
stud@debian:~$ python3 new.py
Enter a number:5

1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

24) 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

```
stud@debian:~$ python3 new.py
Enter a string:fisat
f       : 1
i       : 1
s       : 1
a       : 1
t       : 1
```

25) 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

```
stud@debian:~$ python3 new.py
Enter a string:com
coming
stud@debian:~$ python3 new.py
Enter a string:writing
writingly
stud@debian:~$
```

26) 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

```
Enter the range:6
Enter the words:
ashna
sherin
sulfi
elsa
sudhu
surya
Length of longest word is 6
```

27) 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

```
stud@debian:~$ python3 new.py
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

28) 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

```
stud@debian:~$ python3 new.py
Enter a number:30
Factors are
1
2
3
5
6
10
15
30
stud@debian:~$
```

COURSE OUTCOME 3

29) 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(1
```

Output

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd Desktop

C:\Users\ASUS\Desktop>cd python

C:\Users\ASUS\Desktop\python>md Graphics1

C:\Users\ASUS\Desktop\python>cd graphics1

C:\Users\ASUS\Desktop\python\Graphics1>notepad circle.py

C:\Users\ASUS\Desktop\python\Graphics1>notepad rectangle.py

C:\Users\ASUS\Desktop\python\Graphics1>md tdgraphics

C:\Users\ASUS\Desktop\python\Graphics1>cd tdgraphics

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>notepad cuboid.py

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>notepad sphere.py

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>cd..

C:\Users\ASUS\Desktop\python\Graphics1>cd..
```

```
C:\Users\ASUS\Desktop\python>notepad driver1.py

C:\Users\ASUS\Desktop\python>python driver1.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

C:\Users\ASUS\Desktop\python>
```

COURSE OUTCOME 4

30) 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

```
Enter length of rectangle1: 8
Enter breadth of rectangle1: 6
Area: 48
Perimeter: 28
Enter length of rectangle2: 6
Enter breadth of rectangle2: 4
Area: 24
Perimeter: 20
First rectangle is larger
```

31) 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,type_of_acc,balance):
self.acc_no= acc_no
self.name=name
self.type_of_acc=type_of_acc
self.balance=balance
def deposit(self,x):
self.balance=self.balance+x
print("balance after deposit is=",self.balance)
def withdraw(self,y):
```

```
self.balance=self.balance-y
print("balance after withdrawal is=",self.balance)
x=int(input("amount to be deposited"))
y=int(input("amount to withdraw"))
ob1=Bank(1,"aaa","ccc",300000)
ob2=Bank(2,"bbb","ccc",500000)
ob1.deposit(x)
ob1.withdraw(y)
ob2.deposit(x)
 ob2.withdraw(y)
```

Output

```
amount to be deposited20000
amount to withdraw30000
balance after deposit is= 320000
balance after withdrawal is= 290000
balance after deposit is= 520000
balance after withdrawal is= 490000
```

---

32) 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 area(self):
return self.length * self.breadth
def perimeter(self):
return 2*(self.length + self.breadth)
def __lt__(self,rr):
if (self.length >rr .length and self.breadth > rr.breadth):
print("Area of first rectangle is greater")
else:
        print("Area of second rectangle is greater")
c=int(input("enter length of 1st rectangle"))
d=int(input("enter breadth 1st rectangle"))
u=int(input("enter length of 2nd rectangle"))
v=int(input("enter breadth of 2nd rectangle"))
r1= Rectangle(c,d)
```

```
r3= Rectangle(u,v)
a=r1.area()
b=r3.area()
print("area of 1st rectangle is:",a)
print("perimeter is:",r1.perimeter())
print("area of 2nd rectangle is:",b)
        r1 < r3
```

output

```
enter length of 1st rectangle2
enter breadth 1st rectangle3
enter length of 2nd rectangle1
enter breadth of 2nd rectangle5
area of 1st rectangle is: 6
perimeter is: 10
area of 2nd rectangle is: 5
Area of second rectangle is greater
```

33) Create a class Time with private attributes hour, minute and second.

Overload '+' operator to find sum of 2 time.

Source code

```
Program code:
class Time:
def __init__(self,hr,min,sec):
self.hr=hr
self.min=min
self.sec=sec
def __add__ (self,t):
return(self.hr+t.hr,self.min+t.min,self.sec+t.sec)
t1=Time(3,20,35)
t2=Time(2,25,40)
        print(t1+t2)
```

Output

```
(5, 45, 75)
```

34) 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):
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("XYZ Publications","Wings of Fire","APJ ABDUL
KALAM",100,500)
p.display3()
```

Output

```
Wings of Fire
APJ ABDUL KALAM
Wings of Fire
APJ ABDUL KALAM
100
500
```

COURSE OUTCOME-5

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

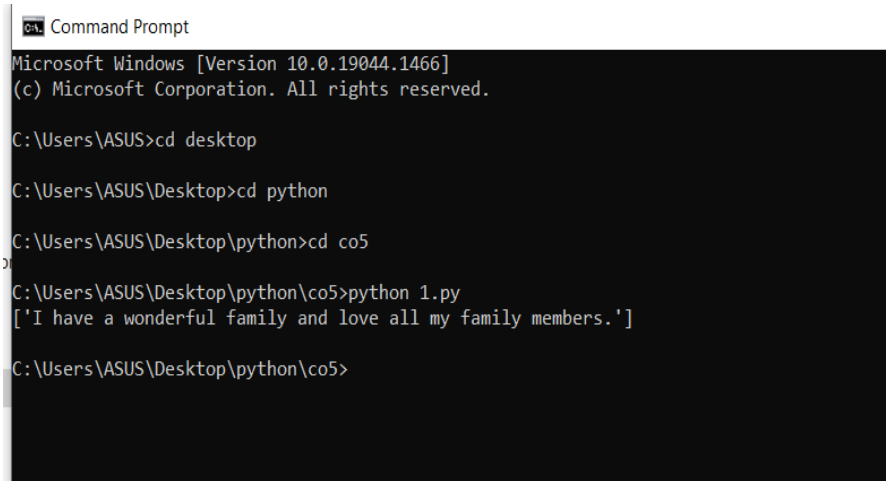
Source code

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

text.txt

```
I have a wonderful family and love all my family members.
```

Output



36) 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('work.csv', 'r') as file:
   reader = csv.reader(file)
   for row in reader:
      print(row)
```

work.csv

	A	B	C	D
1	Name	Age	Profession	
2	Das	40	Manager	
3	Vinu	38	Ass.Manager	
4	Manu	35	Staff	
5	Janaki	30	Nurse	
6				

Output