FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB

LABORATORY RECORD

Name: ALBIN SEBANA KURIAN

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: ARoll No: 13 Reg No: FIT21MCA-2013

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 ***ALBIN SEBANA KURIAN(FIT21MCA-2013)*** *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

9	**10/11/21**	Accept the radius from user and find area of circle.	7	
10	**11/10/21**	Find biggest of 3 numbers entered.	8	
11	**11/10/21**	Accept a file name from user and print extension of that.	8	
12	**11/10/21**	Create a list of colors from commaseparated color names entered by user. Display first and last colors.	9	
13	**11/10/21**	Accept an integer n and compute n+nn+nnn.	9	
14	**11/10/21**	Print out all colors from color-list1 not contained in color-list2.	9	
15	**17/11/21**	Create a single string separated with space from two strings by swapping the character at position 1.	**10**	
16	**17/11/21**	Merge two dictionaries.	**10**	
17	**17/11/21**	Find gcd of 2 numbers.	**11**	
18	**17/11/21**	From a list of integers, create a list removing even numbers.	**12**	
CO 2				
1	**25/11/21**	Program to find the factorial of a number	**13**	
2	**25/11/21**	Generate Fibonacci series of N terms	**13**	

3	**25/11/21**	Find the sum of all items in a list	**14**	
4	**25/11/21**	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	**14**	
5	**02/12/21**	Display the given pyramid with step number accepted from user.	**15**	
6	**02/12/21**	Count the number of characters (character frequency) in a string.	**15**	
7	**02/12/21**	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.	**16**	
8	**09/12/21**	Accept a list of words and return length of longest word.	**16**	
9	**09/12/21**	Construct following pattern using nested loop. *	**17**	
10	**09/12/21**	Generate all factors of a number.	18	
CO 3				

1	**29/01/22**	Create a package graphics with modules rectangle, circle and sub-package 3Dgraphics with modules cuboid and sphere. Include methods to find area and perimeter. Write programs that finds area and perimeter of figures by different importing statements.	19	
CO 4				
1	**13/01/22**	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two	21	
		Rectangle objects by their area.		
2	**13/01/22**	Create a Bank account with members acc number, name, type of acc and balance. Write constructor , methods to deposit at the bank ,withdraw amount from the bank.	**22**	
3	**13/01/22**	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	**24**	
4	**20/01/22**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	**25**	

5	**20/01/22**	Create a class Publisher . Derive class Book from Publisher with attributes title, 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, method overriding.	27	
CO 5				
1	**03/02/22**	Write a Python program to read a file line by line and store it into a list.	29	
2	**03/02/22**	Write a Python program to read each row from a given csv file and print a list of strings.	30	

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"); y=int(input('enter the year')) print('leap
years') for y in range(2021,y+1): if(y % 4 == 0):
print(y)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
print leap year between two given years
enter the year2050
leap years
2024
2028
2032
2036
2040
2044
2048
```

2) List comprehensions:

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

Source code

```
print('a. Generate positive list of numbers from a given
list of intergers') a=[12,-5,-6,-4,11,33,66] print('positive
intergers') for i in a:  if i>=0:
 print(i)
```

b. Square of N numbers

Source code

```
print('b. Square of N
numbers') b=[2,3,5]
```

```
print('Square of numbers') for
i in b:  i=i*i
 print(i)
```

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

Source code

```
print('c. Form a list of vowels selected from a given word')
c = input("Enter any statement : ") vowel =['a','e','i','o','u']
li=[] for i in c:
 if (i in vowel and i not in li):
 li.append(i) print("Vowels present in given
statement : ",li)
```

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

Source code

```
print('d. List ordinal value of each element') d =
input("Enter any statement : ") print("The ASCII
value of '" + d + "' is", ord(d))
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
a. Generate positive list of numbers from a given list of intergers
positive intergers
15
34
51
23
82
b. Square of N numbers
Square of numbers
49
81
64
c. Form a list of vowels selected from a given word
Enter any statement : computer language
Vowels present in given statement :  ['o', 'u', 'e', 'a']
d. List ordinal value of each element
Enter any statement : Z
The ASCII value of 'Z' is 90
```

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

Source code

```
str=input("enter the string:
") counts = dict() words =
str.split() for i in words:  if i
in counts:
 counts[i] += 1
 else:
counts[i] = 1
print("count")
print(counts)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the string: better late than never late
count:
{'better': 1, 'late': 2, 'than': 1, 'never': 1}
```

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

Source code

```
n = int(input('enter the size: ')) lt = [] for
i in range(0,n):
 x=int(input())
 if(x>=100):
 lt.append('OVER')
 else:
    lt.append(x)print(it)
```

Out put

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the size: 5
2
4
68
0
5
[2, 4, 68, 0, 5]
```

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

Source code

```
l = ['Jan','may','mar']
count=0
for i in l:
 num= i.count('a')
count=count+num
print(count)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
['jan', 'may', 'mar']
3
```

6) Enter 2 lists of integers.Check

a. whether list are of same length

b. whether list sums of same value

c. whether any value occur in both.

Source code

```
L1=[1,2,3,4,5]
L2=[1,2,34,4,5]
L3=[] print(L1)
print(L2)
sum1=0
sum2=0
f=0 print("a.Whether list are of same
size") if len(L1)==len(L2): print("The list
are of same size") else: print("The list are
not of same size") print("b.Whether list
```

```
sums to same value") for i in
range(len(L1)): sum1=sum1+L1[i]
for i in range(len(L2)): sum2=sum2+L2[i]
if(sum1==sum2): print("The two lists are of same
value") else: print("The two lists are not of same
value") print("c.Whether any value occur in both
list") for i in L1: if(i in L2): print(i,"the element
occur in both the list") f=1 if(f==0): print("There
are no element occur in both the list")
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
[22, 33, 44, 55, 66]
[77, 88, 99, 22]
a.same size
not same size
b.list sums to same value
not same value
c.Whether any value occur in both list
22 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]

Source code

```
str1=input('enter the string:
') char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print(str1)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the string: albinsebana
albinseb$n$
```

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

[eg:python->nythop]

Source code

```
s=input("enter a
string:") print("original
string:",s) sf=s[0] sl=s[-1]
n=len(s) ns=sl+s[1:n-
1]+sf print(ns)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter string:albin
nlbia
```

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

Source code

```
y=int(input('enter the radius: '))
r=3.14*y*y
print("Area of circle: ",r)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the radius: 56
Area of circle:  9847.04
```

10)Find the biggest of 3 numbers

Source code

```
print("Enter the three numbers:
") a=int(input()) b=int(input())
c=int(input()) if a>b and a>c: print("The
biggest of three numbers: ",a) if b>a
and b>c: print("The biggest of three
numbers: ",b) if c>a and c>b:
print("The biggest of three numbers: ",c)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
three numbers:
76
35
54
biggest  76
```

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:~/albin_13/python$ python3 prg.py
Enter file nameprg.py
The extension of file prg.py is ('prg', '.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:~/albin_13/python$ python3 prg.py
enter the size: 7
enter the color:
pink
black
white
blue
green
orange
yellow
The first color:  pink
The last color:  yellow
```

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

Source code

```
i=int(input("Enter a number:"))
a=i*1 b=i*11 c=i*111
print(a+b+c)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
Enter a number:8
984
```

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

Source code

```
l1=["red","green","blue"]
l2=["green","black","white"]
print(l1) print(l2) for i in l1:
if i not in l2: print(i)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
['yellow', 'green', 'black', 'red', 'violet']
['lemonda', 'blue', 'maroon', 'yellow']
green
black
red
violet
```

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

Source code

```
a=input("enter string 1:")
b=input("enter string 2:")
new_a = b[:1] + a[1:]  new_b
= a[:1] + b[1:]  c=new_a+ ' ' +
new_b  print(c)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter string 1:ALBIN
enter string 2:SEBANA
SLBIN AEBANA
```

16)Merge two dictionaries.

Source code

```
thisdict ={  "house
no": "345",
 "gender": “M”,
 "dob": "25/05/1998"
}
Dic={"name":'Albin Sebana' , "phoneno":'123456789'}
Dic.update(thisdict) print(Dic)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
{'name': 'Albin Sebana', 'phoneno': '123456789', 'house no': '345', 'gender': 'M', 'dob': '25/05/1998'}
```

17) Find gcd of 2 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):
gcd=i

print("The gcd of two number is : ",gcd)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter first number 56
enter second number 78
GCD:  2
```

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:~/albin_13/python$ python3 prg.py
[2, 3, 4, 5, 6, 7, 8]
List of even numbers:
[3, 5, 7]
```

COURSE OUTCOME 2

1. Program to find the factorial of a number.

Source code

```
n=int(input('enter the
    number: ')) fact=1
    print("Factorial of the
    number:") for i in range(1,n+1):
    fact=fact*i print(fact)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the number: 7
Factorial of the number:
5040
```

2. Generate fibonacci series of N terms.

Source code

```
n=int(input('Enter the number: '))
a=0 b=1 c=0
print("Fibonacci Series:")
print(a) print(b) for i in
range(3,n+1):
  c=a+b
  print(c)
  a=b b=c
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
Enter the number: 8
Fibonacci Series:
0
1
1
2
3
5
8
13
```

3. Find the sum of all items in a list.

Source code

```
def sum_of_list(l):
 total = 0
 for val in l:
 total = total + val
return total

my_list = [3,5,7,9,2]
print("The sum of my_list is", sum_of_list(my_list)) Output
```

```
stud@debian:~/albin_13/python$ python3 prg.py
The sum of my_list is 45
```

4. 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=1234 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
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
78
80
92
[6084, 6400, 8464]
```

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

Source code

```
for i in range(1,5): for
j in range(1,i+1):
print(i*j,end="")
print("\n")
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
1

24

369

481216
```

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

Source code

```
string = "characters in a st";
count = 0;  for i in range(0,
len(string)):
 if(string[i] != ' '):   count = count + 1;  print("Total number
of characters in a string: " + str(count));
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
Total number of characters in a string: 18
```

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

Source code

```
str1=input("enter a
string:") if str1[-3:]=='ing':
str1=str1+'ly'  else:
 str1=str1+'ing'
print("changed string:",str1)
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter a string:python
changed string: pythoning
stud@debian:~/albin_13/python$ python3 prg.py
enter a string:sleeping
changed string: sleepingly
```

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

Source code

```
list1=[] n=int(input("Enter the
range:")) print("Enter the
words:") for i in range(0,n):
list1.append(input(""))
longest=list1[0] for i in
range(1,n):
if(len(list1[i])>len(longest)):
longest=list1[i] print("Length
of longest word
is",len(longest))
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
Enter the range:4
Enter the words:
python
java
android
shell
Length of longest word is 7
```

9. Construct following pattern using nested loop.

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

Source code

```
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(0,i):
print("*",end="")
print("\n")
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py

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

10. Generate all factors of a number.

Source code

```
n=int(input( "enter
        the number: ")) i=2
        print("the factors of
        ",n) while i <= n : if (n %
        i==0) : print(i) i = i + 1
```

Output

```
stud@debian:~/albin_13/python$ python3 prg.py
enter the number: 123
the factors of  123
3
41
123
```

COURSE OUTCOME 3

1. Create a package graphics with modules rectangle, circle and sub-package 3Dgraphics 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
```

```
stud@debian:~/albin13/python$ mkdir graphics
stud@debian:~/albin13/python$ cd graphics
stud@debian:~/albin13/python/graphics$ gedit circle.py
stud@debian:~/albin13/python/graphics$ gedit circle.py
stud@debian:~/albin13/python/graphics$ gedit rectangle.py
stud@debian:~/albin13/python/graphics$ mkdir tdgraphics
stud@debian:~/albin13/python/graphics$ cd tdgraphics
stud@debian:~/albin13/python/graphics/tdgraphics$ gedit cuboid.py
stud@debian:~/albin13/python/graphics/tdgraphics$ gedit sphere.py
stud@debian:~/albin13/python/graphics/tdgraphics$ cd ..
stud@debian:~/albin13/python/graphics$ cd ..
stud@debian:~/albin13/python$ gedit drive.py
```

```
stud@debian:~/albin13/python$ python3 drive.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

1. 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

```
stud@debian:~/albin_13/python$ python3 prg.py
Enter length of rectangle1: 55
Enter breadth of rectangle1: 45
Area: 2475
Perimeter: 200
Enter length of rectangle2: 33
Enter breadth of rectangle2: 47
Area: 1551
Perimeter: 160
First rectangle is larger
```

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

   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

```
stud@debian:~/albin_13/python$ python3 prg.py
1.deposit
2.withdraw
3.exit

enter your choice: 1
enter the amount to deposit: 10000
balance is : 11000
1.deposit
2.withdraw
3.exit

enter your choice: 2
enter your amount to withdraw: 500
balance is : 10500
1.deposit
2.withdraw
3.exit
```

3. 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

```
stud@debian:~/albin_13/python$ python3 prg.py
Enter the length of the rectangle1: 15
Enter the breadth of the rectangle1: 25
Enter the length of the rectangle2: 13
Enter the breadth of the rectangle2: 26
The area of 1st rectangle:  375
The area of 2nd rectangle:  338
Rectangle 2 is greater
```

4. 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.__mi

n sec=t1.__sec+t2.__sec

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

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

t1+t2
```

Output

```
stud@debian:~/albin_13/python$ python3 prgg.py
7 : 65 : 59
stud@debian:~/albin_13/python$
```

5. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding. 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,aut
hor,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

```
stud@debian:~/albin_13/python$ python3 prgg.py
Taming Python
jeeva jose
100
500
Java programming
E Balagurusami
500
1200
```

COURSE OUTCOME 5

1. 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

```
stud@debian:~/albin_13/python$ python3 prgg.py
['Store a list of first names. Count the occurrences of ‘a’ within the list', 'S
tore a list of first names. Count the occurrences of ‘a’ within the list']
```

2. **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

```
stud@debian:~/albin_13/python$ python3 prgg.py
['java', 'python', 'web']
['44', '56', '34']
```