FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: AMAL V S

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: A Roll No: 15

Register Number: FIT21MCA-2015

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 **AMAL V S (FIT21MCA-2015)** 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 OD

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	28-10-2021	Display future leap years from current year to a final year entered by user.	1	
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)	1-3	
3	28-10-2021	Count the occurrences of each word in a line of text.	3-4	
4	28-10-2021	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.	4	
5	10-11-2021	Store a list of first names. Count the occurrences of 'a' within the list	5	
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	5-6	
7	10-11-2021	Get a string from an input string where all occurrences of first character replaced with '$', except first character. [eg: onion ->oni$n]	7	
8	10-11-2021	Create a string from given string where first and last characters exchanged. [eg: python - >nythop]	7-8	
9	10-11-2021	Accept the radius from user and find area of circle.	8	
10	11-11-2021	Find biggest of 3 numbers entered.	8-9	
11	11-11-2021	Accept a file name from user and print extension of that.	9	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
12	11-11-2021	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	9-10	
13	11-11-2021	Accept an integer n and compute n+nn+nnn.	10	
14	11-11-2021	Print out all colors from color-list1 not contained in color-list2.	10	
15	17-11-2021	Create a single string separated with space from two strings by swapping the character at position 1.	11	
16	17-11-2021	Sort dictionary in ascending and descending order.	11-12	
17	17-11-2021	Merge two dictionaries.	12	
18	17-11-2021	Find gcd of 2 numbers.	12-13	
19	17-11-2021	From a list of integers, create a list removing even numbers.	13	
20	25-11-2021	Program to find the factorial of a number.	14-15	
21	25-11-2021	Generate Fibonacci series of N terms.	15-16	
22	25-11-2021	Find the sum of all items in a list	15	
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.	16	
24	02-12-2021	Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 8 12 16	17	
25	02-12-2021	Count the number of characters (character frequency) in a string.	17-18	
26	02-12-2021	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’	18	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
27	09-12-2021	Accept a list of words and return length of longest word.	18-19	
28	09-12-2021	Construct following pattern using nested loop *	19-20	
29	09-12-2021	Generate all factors of a number.	20	
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)	21-23	
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.	24-25	
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.	26-27	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
33	13-01-2022	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	28-29	
34	20-01-2022	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	29-30	
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.	30-31	
36	03-02-2022	Write a Python program to read a file line by line and store it into a list.	32	
37	03-02-2022	Write a Python program to read each row from a given csv file and print a list of string.	32	

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
print leap year between two given years
Enter end year : 2040
list of leap years
2024
2028
2032
2036
stud@debian:~/Amal V S/Python$
```

2) List comprehensions:

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

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Elements in the list are: [-11, 4, 8, -34, 10, 14]
Positive numbers in the list
4
8
10
14
stud@debian:~/Amal V S/Python$
```

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:~/Amal V S/Python$ python3 Program.py
Enter range:6
1
4
9
16
25
36
stud@debian:~/Amal V S/Python$
```

c. Form a list of vowels selected from a givenword.

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:~/Amal V S/Python$ python3 Program.py
Enter a string: malayalam
vowels in the list are:
['a', 'a', 'a', 'a']
stud@debian:~/Amal V S/Python$
```

d. List ordinal values of each element of aword.

Source code

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

Output

```
stud@debian:~/Amal V S/Python$ python3 Program.py
String: Welcome
Ordinal Values
87
101
108
99
111
109
101
stud@debian:~/Amal V S/Python$
```

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

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:~/Amal V S/Python$ python3 Program.py
Enter a line of text:jack is a good person with good heart
jack     1
is       1
a        1
good     2
person   1
with     1
heart    1
stud@debian:~/Amal V S/Python$
```

4) Prompt the user for a list of integers. For all values greaterthan 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:~/Amal V S/Python$ python3 Program.py
Enter an integer: 5
Enter an integer: 2
Enter an integer: 6
Enter an integer: 8
Enter an integer: 2
Enter an integer: 4
Enter an integer: 5
Enter an integer: 2005
[5, 2, 6, 8, 2, 4, 5, 'over']
Enter an integer: 45
Enter an integer: 78
Enter an integer: 124
[5, 2, 6, 8, 2, 4, 5, 'over', 45, 78, 'over']
Enter an integer:
```

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

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Elements in the list are:
['Amal', 'Ajay', 'akhil', 'sanju']
count of 'a' is: 4
stud@debian:~/Amal V S/Python$
```

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 inboth.

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:
 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

```
stud@debian:~/Amal V S/Python$ python3 Program.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
stud@debian:~/Amal V S/Python$
```

7) Get a string from a input string where all occurrence of first character replacedwith '$',except firstcharacter.[eg:onion->oni$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:~/Amal V S/Python$ python3 Program.py
Enter a string: occupation
Original string is:  occupation
String:  occupati$n
stud@debian:~/Amal V S/Python$
```

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:~/Amal V S/Python$ python3 Program.py
Enter a string: gangster
rangsteg
stud@debian:~/Amal V S/Python$
```

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

Source code

```
r=int(input('Enter the radius: '))
A=3.14*r*r
print(A)
```

Output

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter the radius: 5
Area is 78.5
stud@debian:~/Amal V S/Python$
```

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:~/Amal V S/Python$ python3 Program.py
Enter first number:5
Enter second number:8
Enter third number:2
8
stud@debian:~/Amal V S/Python$
```

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

Source code

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

Output

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter file name:program.txt
The extension of file program.txt is ('program', '.txt')
stud@debian:~/Amal V S/Python$
```

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:~/Amal V S/Python$ python3 Program.py
Enter color names:yellow,orange,red,white,black
['yellow', 'orange', 'red', 'white', 'black']
first color: yellow Last color: black
stud@debian:~/Amal V S/Python$
```

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:~/Amal V S/Python$ python3 Program.py
Enter the number:5
5 + 5 * 5 + 5 * 5 * 5 = 615
stud@debian:~/Amal V S/Python$
```

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

Source code

```
l1=['red','blue','black']
l2=['red',white,pink]
print(l1)
print(l2)
print("Colors that are not in l1:
")
for i in l1:
 if i not in l2:
        print(i)
```

Output

```
colours not in l2 is:
 ['blue', 'black']
```

15) Create a single string separated with space from two strings by swapping the character at position1.

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:~/Amal V S/Python$ python3 Program.py
Enter first string:amal
Enter second string:vs
vmal as
stud@debian:~/Amal V S/Python$
```

16) Sort dictionary 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)
```

output

```
[('d', 2), ('c', 3), ('a', 1), ('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 twodictionaries.

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

```
stud@debian:~/Amal V S/Python$ python3 Program.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'}
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter first number: 5
Enter first number: 60
GCD is 5
stud@debian:~/Amal V S/Python$
```

19) From a list of integers, create a list removing evennumbers.

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

COURSE OUTCOME 2

20) Program to find the factorial of anumber.

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:~/Amal V S/Python$ python3 Program.py
Enter a number:6
720
stud@debian:~/Amal V S/Python$
```

21) Generate fibonacci series of Nterms.

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:~/Amal V S/Python$ python3 Program.py
Enter a limit:10
0
1
1
2
3
5
8
13
21
34
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
List elements are: [2, 6, 9, 11, 25]
The sum of list elements is: 53
stud@debian:~/Amal V S/Python$
```

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

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:~/Amal V S/Python$ python3 Program.py
68
78
80
92
[4624, 6084, 6400, 8464]
stud@debian:~/Amal V S/Python$
```

24) Display the given pyramid with step number accepted fromuser.

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:~/Amal V S/Python$ python3 Program.py
Enter a number:6


1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

stud@debian:~/Amal V S/Python$
```

25) Count the number of characters (character frequency) in astring.

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:~/Amal V S/Python$ python3 Program.py
Enter a string:amal
a       : 2
m       : 1
l       : 1
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter a string:walk
walking
stud@debian:~/Amal V S/Python$
```

27) Accept a list of words and return length of longestword.

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter the range:4
Enter the words:
hello
guys
welcome
to
Length of longest word is 7
stud@debian:~/Amal V S/Python$
```

28) Construct following pattern using nestedloop.

```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

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:~/Amal V S/Python$ python3 Program.py
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*

stud@debian:~/Amal V S/Python$
```

29) Generate all factors of anumber.

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:~/Amal V S/Python$ python3 Program.py
Enter a number:20
Factors are
1
2
4
5
10
20
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ mkdir graphics
stud@debian:~/Amal V S/Python$ cd graphics
stud@debian:~/Amal V S/Python/graphics$ gedit circle.py
stud@debian:~/Amal V S/Python/graphics$ gedit rectangle.py
stud@debian:~/Amal V S/Python/graphics$ mkdir tdgraphics
stud@debian:~/Amal V S/Python/graphics$ cd tdgraphics
stud@debian:~/Amal V S/Python/graphics/tdgraphics$ gedit cuboid.py
stud@debian:~/Amal V S/Python/graphics/tdgraphics$ gedit sphere.py
stud@debian:~/Amal V S/Python/graphics/tdgraphics$ cd ..
stud@debian:~/Amal V S/Python/graphics$ cd ..
stud@debian:~/Amal V S/Python$ gedit drive.py
stud@debian:~/Amal V S/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
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter length of rectangle1: 6
Enter breadth of rectangle1: 4
Area: 24
Perimeter: 20
Enter length of rectangle2: 7
Enter breadth of rectangle2: 8
Area: 56
Perimeter: 30
Second rectangle is larger
stud@debian:~/Amal V S/Python$
```

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):
        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)
```

```
elifch==2:
        amt=int(input("Enter the amount:"))
        b=acc1.withdraw(amt)
        print("Current balance:",b)
elifch==3:
        cb=acc1.display_balance()
        print("Current balance:",cb)
elifch==4:
        exit(1)
else:
        print("Invalid choice")
```

Output

```
stud@debian:~/Amal V S/Python$ python3 Program.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:3000
Current balance: 49000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:4
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Enter length of rectangle1: 5
Enter breadth of rectangle1: 3
Enter length of rectangle2: 8
Enter breadth of rectangle2: 5
Second rectangle is larger
stud@debian:~/Amal V S/Python$
```

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

Output

```
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):
        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","Javaprogramming","E Balagurusami",500,1200)
q.display3()
```

Output

```
stud@debian:~/Amal V S/Python$ python3 Program.py
Taming Python
jeeva jose
100
500
Java programming
E Balagurusami
500
1200
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python$ python3 Program.py
['Aluva also known by its former name Alwaye is a region in the city of Kochi in
 Kerala, India. It is also a part of the Kochi metropolitan area and is situated
 around 15 km from the city center on the banks of Periyar River. A major transp
ortation hub, with easy access to all major forms of transportation, Aluva acts
as a corridor which links the highland districts to the rest of Kerala. Cochin I
nternational Airport at Nedumbassery is 11.7 km from Aluva. Aluva is accessible
through rail , air , metro  along with major highways and roadlines. Aluva KSRTC
 bus station is an important transport hub in Kerala and one of the busiest stat
ions in central part of the state.']
stud@debian:~/Amal V S/Python$
```

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

```
stud@debian:~/Amal V S/Python/new$ python3 Hack.py
['Name', 'age', 'Job ']
['Amal', '22', 'Designer']
['Ajay', '30', 'Sales Manager']
['Rahul', '25', 'Accountant']
stud@debian:~/Amal V S/Python/new$
```