FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ELIZABA MARIYAM BINNY

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: B** **Roll No: 53**

Register Number: ***FIT21MCA-2053***

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 ***Elizaba Mariyam Binny FIT21MCA-2053*** *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

SI No:	Date:	Name of Experiment:	Page No:	Signature of Staff –In – Charge:
1	28-10-21	Display future leap years from current year to a final year entered by user.	1	
2	28-10-21	List comprehensions:	1	
3	28-10-21	Count the occurrences of each word in a line of text.	2	
4	28-10-21	Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.	3	
5	10-11-21	Store a list of first names. Count the occurrences of ‘a’ within the list	3	
6	10-11-21	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	4	
7	10-11-21	Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.	5	
8	10-11-21	Create a string from given string where first and last characters exchanged.	5	
9	10-11-21	Accept the radius from user and find area of circle.	6	
10	11-11-21	Find biggest of 3 numbers entered.	6	
11	11-11-21	Accept a file name from user and print extension of that.	6	
12	11-11-21	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	7	
13	11-11-21	Accept an integer n and compute n+nn+nnn	7	
14	11-11-21	Print out all colors from color-list1 not contained in color-list2.	7	
15	11-11-21	Create a single string separated with space from two strings by swapping the character at position 1.	8	
16	17-11-21	Sort dictionary in ascending and descending order	8	

17	17-11-21	Merge two dictionaries	9	
18	17-11-21	Find gcd of 2 numbers.	9	
19	17-11-21	From a list of integers, create a list removing even numbers.	9	
20	17-11-21	Program to find the factorial of a number	10	
21	25-11-21	Generate Fibonacci series of N terms	10	
22	25-11-21	Find the sum of all items in a list	11	
23	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.	11	
24	25-11-21	Display a pyramid with step number accepted from user.	12	
25	02-12-21	Count the number of characters (character frequency) in a string.	12	
26	02-12-21	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.	13	
27	02-12-21	Accept a list of words and return length of longest word.	13	
28	09-12-21	Construct a pattern using nested loop	14	
29	09-12-21	Generate all factors of a number.	15	
30	09-12-21	Write lambda functions to find area of square, rectangle and triangle.	15	
31	13-12-21	Work with built-in packages 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	16	
32	13-12-21	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	18	

33	13-12-21	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.	18	
34	20-01-22	Create a class Rectangle with private attributes length and width. Overload ‘	19	
35	20-01-22	Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.	20	
36	03-02-22	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.	21	
37	03-02-22	Write a python program to read a file line by line and store it into a list.	22	
38	03-02-22	Write a python program to read each row from a given csv file and print a list of strings	22	

COURSE OUTCOME 1

1. **Display future leap year from current leap year to a final year entered by user.**

PROGRAM:

```
print ("Enter last year")
endYear = int(input())
from datetime import date
today = date.today()
print ("List of leap years:")
for year in range(today.year, endYear+1):
  if (year % 4==0) and (year % 100 != 0) or (0 == year % 400):
    print (year)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co2.py
Enter last year
2028
List of leap years:
2024
2028
```

2. **List comprehensions:**
   (a) Generate positive list of numbers from a given list of integers
   (b) Square of N numbers
   (c) Form a list of vowels selected from a given word
   (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

PROGRAM:

```
list1 = []
n = int(input("Enter number of elements : "))
print(" enter numbers to find positive : ")
for i in range(0, n):
    element = int(input())
    list1.append(element)
print("positive numbers are : ")
s=[p for p in list1 if p>0]
print(s)

print("square number are : ")
for num in list1:
      print(num*num)

word = input("Enter any word : ")
```

```
vowels =['a','e','i','o','u']
list1=[]
print("Vowels present in given statement : ",[x for x in word if x in vowels and
x not in list1])

word1 = input("Enter any word : ")
print("Ordinal values are : ",[ord(x)for x in word1])
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co3.py
Enter number of elements : 10
 enter numbers to find positive :
-1
0
-2
2
1
30
48
65
34
3
positive numbers are :
[2, 1, 30, 48, 65, 34, 3]
square number are :
1
0
4
4
1
900
2304
4225
1156
9
Enter any word : alphabet
Vowels present in given statement :  ['a', 'a', 'e']
Enter any word : python
Ordinal values are :  [112, 121, 116, 104, 111, 110]
```

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

PROGRAM:

```
str1=input("Enter a string: ")
counts={}

words=str1.split()
for word in words:
        if word in counts:
                counts[word]+=1
        else:
                counts[word]=1
for k,v in counts.items():
        print(k,v)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co4.py
Enter a string: watermelon juice apple juice
watermelon 1
juice 2
apple 1
```

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

PROGRAM

```
list=[]
n=int(input("Enter the number of integers: "))
count=1
for num in range(0,n):
  num=int(input("Enter integer no %d : "%count))
  list.append(num)
  count+=1
print(list)
for m in range(0,n):
  if list[m]>100:
    list[m]="over"
print("List after correction : ",list)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co4.py
Enter the number of integers: 3
Enter integer no 1 : 234
Enter integer no 2 : 23
Enter integer no 3 : 45
[234, 23, 45]
List after correction :  ['over', 23, 45]
```

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

PROGRAM:

```
list1 = []
count=0
n = int(input("Enter number of elements : "))
print(" enter names : ")
for i in range(0, n):
    element = input()
    list1.append(element)
```

```
for i in range(0, n):
        for y in list1[i]:
                if y=="a" or y=="A":
                        count=count+1
print("Number of a is : ",count)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co6.py
Enter number of elements : 4
 enter names :
anagha
john
max
jerry
Number of a is :  4
```

6. **Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both**

PROGRAM

```
l1=[1,2,3,4]
l2=[7,8]
s=0
q=0
c=0
l=0
flag=0
for i in l1:
        s=s+i
        c=c+1
        if i in l2:
                print(i,"occurs in both list")
                flag=1
if flag==0:
        print("no common values")
for j in l2:
        q=q+j
        l=l+1
print (s,q)
if(c==l):
        print("same length both are",c)
else:
        print("different, length of list is ",c,"and list 2 is",l)
if(s==q):
        print("same sum both are",s)
else: print("different, sum of list 1 is",s,"and list 2 is",q)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co7.py
no common values
10 15
different, length of list is  4 and list 2 is 2
different, sum of list 1 is 10 and list 2 is 15
```

7. **Get a string from an input string where all occurrences of first character replaced with '$', except first character.**

PROGRAM

```
strl=input("enter string:")
print("original string",strl)
char=strl[0]
strl=strl.replace(char,'$')
strl=char+strl[1:]
print("replaced string:",strl)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co8.py
enter string:starts
original string starts
replaced string: start$
```

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

PROGRAM

```
strl=input("enter string:")
n=len(strl)
print("original string",strl)
charf=strl[0]
charl=strl[n-1]
strl=charl+strl[1:n-1]+charf
print("replaced string:",strl)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co9.py
enter string:india is my country
original string india is my country
replaced string: yndia is my countri
```

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

PROGRAM

```
r = int(input ('enter radius'))
a = 3.14*r*r
print (a)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co10.py
enter radius10
314.0
```

10. Find biggest of 3 numbers entered.

PROGRAM

```
a= int(input ('enter a number'))
b= int(input ('enter a number'))
c= int(input ('enter a number'))
if a>b:
  if a>c:
    print("the greatest number is",a)
  else:
    print("the greatest number is",c)
else:
  if b>c:
    print("the greatest number is",b)
  else:
    print("the greatest number is",c)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co11.py
enter a number2
enter a number8
enter a number6
the greatest number is 8
```

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

PROGRAM

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

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co12.py
enter a file name 1co12.py
The extension of  1co12.py is (' 1co12', '.py')
```

12. Create a list of colors from comma-separated color names entered by user.
Display first and last colors.

PROGRAM

```
list1 =[]
a= list(map(str,input("Enter values\n").split(',')))
print("\nThe values of input are", a)
print(min(a))
print(max(a))
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co13.py
Enter values
black,blue,yellow,green

The values of input are ['black', 'blue', 'yellow', 'green']
black
yellow
```

13. Accept an integer n and compute n+nn+nun.

PROGRAM

```
n=int(input("enter a number"))
print(n*123)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co15.py
blue does not occur in both list
```

14. Print out all colors from color-list I not contained in color-list2.

PROGRAM

```
l1=['red','blue','green']
l2=['red','blue','green']
flag=0
for i in l1:
```

```
        if i not in l2:
                print(i,"does not occur in both list")
                flag=1
if flag==0:
        print("common values only")
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co15.py
blue does not occur in both list
```

15. Create a single string seperated with space from two strings by swapping the character at position one.

PROGRAM

```
str1=input("Enter first string:")
str2=input("Enter second string:")
str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]
print(str3)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co16.py
Enter first string:python
Enter second string:java
jython pava
```

16. Sort dictionary in ascending and descending order.

PROGRAM

```
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

```
stud@debian:~/elizabamca/python$ python3 1c017.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 2 dictionary

PROGRAM

```
dict1={"Name":"Athidhi","Age":25}
dict2={"Gender":"F","Qualification":"PG"}
dict1.update(dict2)
print(dict1)
print(dict2)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co18.py
{'Name': 'Athidhi', 'Age': 25, 'Gender': 'F', 'Qualification': 'PG'}
{'Gender': 'F', 'Qualification': 'PG'}
```

18. Find gcd of 2 numbers.

PROGRAM

```
num1=int(input("enter two numbers"))
num2=int(input("enter two numbers"))
small=min(num1,num2)
for i in range(1,small+1):
        if(num1%i==0 and num2%i==0):
                j=i
                print("factor is",j)
print("GCD is",j)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co19.py
enter two numbers23
enter two numbers46
factor is 1
factor is 23
GCD is 23
```

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

PROGRAM

```
l1=[1,2,3,4,5,6,7,8,9]
l2=[]
for i in l1:
        if i%2==1:
                l2.append(i)
print(l2)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co20.py
[1, 3, 5, 7, 9]
```

COURSE OUTCOME 2

20. Program to find the factorial of a number

PROGRAM

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

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co1.py
enter a number5
120
```

21. Generate Fibonacci series of N terms

PROGRAM

```
a= int(input ('enter a number'))
f1=0
f2=1
print(f1)
print(f2)
for i in (range( 0,a-2 )):
  f3=f1+f2
  print(f3)
  f1=f2
  f2=f3
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co2.py
enter a number10
0
1
1
2
3
5
8
13
21
34
```

22. Find the sum of all items in a list

PROGRAM

```
list=[]
n=int(input("Enter the number of integers: "))
count=1
for num in range(0,n):
 num=int(input("Enter integer no %d : "%count))
 list.append(num)
count+=1
print("List of numbers are ",list)
sum=0
for num in list:
 sum =sum+num
print("Sum of the numbers in the list are =",sum)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co22.py
Enter the number of integers: 4
Enter integer no 1 : 1
Enter integer no 1 : 3
Enter integer no 1 : 5
Enter integer no 1 : 7
List of numbers are  [1, 3, 5, 7]
Sum of the numbers in the list are = 16
```

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

PROGRAM

```
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:~/elizabamca/python$ python3 2co4.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

24. Display the given pyramid with step number accepted from user.
Eg: NO=4
1
2 4
3 6 9
4 8 12 16

PROGRAM

```
a=int(input("enter a number"))
for row in range(1,a+1):
        c=row
        for col in range(0,row):
                print(c, end=" ")
                c=c+row
        print("\r")
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co5.py
enter a number4
1
2 4
3 6 9
4 8 12 16
```

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

PROGRAM

```
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:~/elizabamca/python$ python3 2co6.py
Enter a string:java
j       : 1
a       : 2
v       : 1
```

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

PROGRAM

```
a=input("Enter a word\n")
l=len(a)
ll=a[l-3:l]
if(ll=="ing"):
        s=a+"ly"
else:
        s=a+"ing"
print (s)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co7.py
Enter a word
happy
happying
```

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

PROGRAM

```
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:~/elizabamca/python$ python3 2co8.py
Enter the range:4
Enter the words:
python
java
php
html
Length of longest word is 6
```

28. Construct following pattern using nested loop

```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

PROGRAM

```
a=int(input("enter a number"))
for row in range(0,a):
        for col in range(0,row):
                print("*  ", end=" ")
        print("\r")
for row in range(a,0,-1):
        for col in range(row,0,-1):
                print("*  ", end=" ")
        print("\r")
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co9.py
enter a number5

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

29. Generate all factors of a number.

PROGRAM

```
num1=int(input("enter number"))
for i in range(1,num1+1):
        if(num1%i==0):
                j=i
                print("factor is",j)
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 2co10.py
enter number40
factor is 1
factor is 2
factor is 4
factor is 5
factor is 8
factor is 10
factor is 20
factor is 40
```

30. Write lambda functions to find area of square, rectangle and triangle.

PROGRAM

```
import math
triangle=lambda s,p,q,r:(s*(s-p)*(s-q)*(s-r))**0.5
rectangle=lambda l,h:l*h
square=lambda a:a*a
print("Area of triangle (3,4,5) is",triangle(6,3,4,5))
print("Area of rectangle (3,4) is",rectangle(3,4))
print("Area of square (3) is",square(3))
```

OUTPUT

```
stud@debian:~/elizabamca/python$ python3 1co30.py
Area of triangle (3,4,5) is 6.0
Area of rectangle (3,4) is 12
Area of square (3) is 9
```

COURSE OUTCOME 3

31. Work with built-in packages

Create a package graphics with modules rectangle, circle and sub package 3D (td)- graphics with modules cuboid & 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).

PROGRAM

Circle.py

```
from math import pi
def area_circle(radius):
 return pi*radius*radius
def perimeter_circle(radius):
 return 2*pi*radius
```

rectangle.py

```
def area_rec(length,width):
 return length*width
def perimeter_rec(length,width):
 return 2*(length+width)
```

sphere.py

```
from math import pi
def area_sphere(radius):
 return 4*(pi*radius*radius)
def perimeter_sphere(radius):
 return 2*pi*radius
```

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
```

driver1.py

```
import Graphics1
from Graphics1 import circle,rectangle
from Graphics1.tdgraphics import cuboid,sphere
from Graphics1.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

```
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

32. Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

PROGRAM

Program 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)
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)
if (a>b):
print("1st is greater")
else:
        print("2nd is greater")
```

OUTPUT:

```
enter length of 1st rectangle3
enter breadth 1st rectangle4
enter length of 2nd rectangle5
enter breadth of 2nd rectangle7
area of 1st rectangle is: 12
perimeter is: 14
area of 2nd rectangle is: 35
2nd is greater
```

33. 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.

PROGRAM

```
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
```

34. Create a class Rectangle with private attributes length and width. Overload '>' operator to compare the area of 2 rectangles.

PROGRAM

```
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"))
  :("enter breadth of 2nd rectangle"))
  gle(c,d)
  gle(u,v)

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

(5, 45, 75)

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
```

35. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

PROGRAM

```
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(2,25,20)
        print(t1+t2)
```

OUTPUT

```
(5, 45, 55)
```

36. 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.

PROGRAM

```
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

37. Write a python program to read a file line by line and store it into a list.

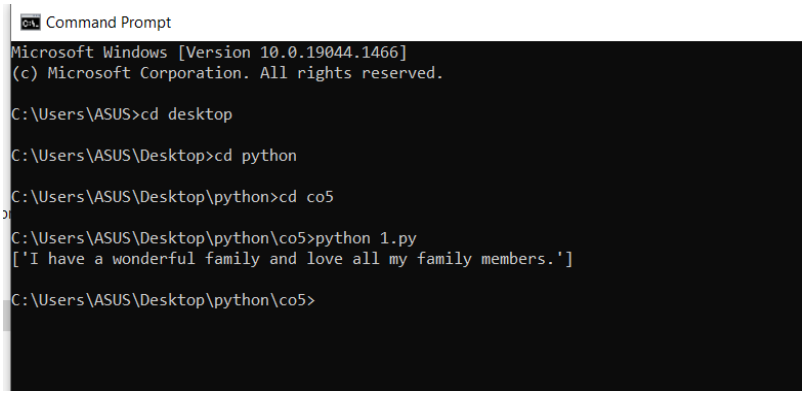
PROGRAM

```
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



38. Write a python program to read each row from a given csv file and print a list of strings

PROGRAM

```
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

```
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>cd co5

C:\Users\ASUS\Desktop\python\co5>python 2.py
['Name', 'Age', 'Profession']
['Das', '40', 'Manager']
['Vinu', '38', 'Ass.Manager']
['Manu', '35', 'Staff']
['Janaki', '30', 'Nurse']

C:\Users\ASUS\Desktop\python\co5>
```