FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

LABORATORY RECORD

20MCA131 - PROGRAMMING LAB

Name : DHANIK DINTO

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 Batch: 2021 A Roll No: 52

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 ***DHANIK DINTO (FIT21MCA-2052)*** *in the* ***20MCA131 PROGRAMMING*** *Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

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

Name: — Name:

Date:

Date of University practical examination ……

Signature of Internal Examiner

Signature of External Examine

CONTENT

12	**11-11-21**	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	**8**	
13	**11-11-21**	Accept an integer n and compute n+nn+nnn	**8**	
14	**11-11-21**	Print out all colors from color-list1 not contained in color-list2.	**9**	
15	**11-11-21**	Create a single string separated with space from two strings by swapping the character at position 1	**9**	
16	**17-11-21**	Sort dictionary in ascending and descending order.	**10**	
17	**17-11-21**	Merge two dictionaries	**10**	
18	**17-11-21**	Find gcd of 2 numbers.	**11**	
19	**17-11-21**	From a list of integers, create a list removing even numbers.	**11**	
20	**17-11-21**	Program to find the factorial of a number	**12**	
21	**25-11-21**	Generate Fibonacci series of N terms.	**12**	
22	**25-11-21**	Find the sum of all items in a list	**13**	
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	**13**	
24	**25-11-21**	Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 8 12 16	**14**	
25	**02-12-21**	Count the number of characters (character frequency) in a string	**15**	

26	**02-12-21**	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’	**16**	
27	**02-12-21**	Accept a list of words and return length of longest word.	**16**	
28	**09-12-21**	Construct following pattern using nested loop *	**17**	
29	**09-12-21**	Generate all factors of a number	**18**	
30	**09-12-21**	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)	**19**	
31	**13-01-22**	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	**21**	
32	**13-01-22**	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	**22**	
33	**13-01-22**	Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.	**23**	
34	**20-01-22**	Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.	**24**	

35	**20-01-22**	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.	**24**	
36	**03-02-22**	Write a Python program to read a file line by line and store it into a list.	**26**	
37	**03-02-22**	Write a Python program to read each row from a given csv file and print a list of string.	**26**	

COURSE OUTCOME 1

PROGRAM-1

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

Program Code:

```
print("Print leap year between two given years")
print("Enter start year")
start Year=int(input())
print("Enter last year")
end Year=int(input())
print("List of leap years:")
for year in range(start Year, end Year):
if(0==year%4)and(0!=year%100)or(0==year%400):
        print(year)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 Leap2.py
Print leap year between two given years
Enter start year
2020
Enter last year
2028
List of leap years:
2020
2024
```

PROGRAM-2

List Comprehensions:

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

        print(num)

Program Code:

```
list=[1,-2,67,45,-5]
for num in list:
if num>0:
print(num)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 pos.py
10
4
35
67
```

b)Square of Nnumbers.

Program code:

```
list=[5,8,-1,-2]
for num in list:
        print(num * num)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 square.py
1
4
9
16
25
36
49
```

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

Program Code:

```
stringA="EmElSha"
print("Given String:\n",stringA)
vowels="AaEeIiOoUu"
li=[]
for r in stringA:
if r in vowels:
li.append(r)
        print(li)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 listvov.py
Enter a string
hello
Given String:
 hello
The vowels present in the string:
  ['e', 'o']
```

d)List ordinal value of each element of a word

.

Program code:

```
stringp="Fisat"
"'for c in stringp:
print(ord(c))"'
s=[ord(p)for p in stringp]
        print(s)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 ord.py
Enter a string
ygss
y
121
g
103
s
115
s
115
```

PROGRAM-3

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

Program code:

```
s="HI hello, i am mia and i am dia"
l=s.split()
d={x:l.count(x) for x in l}
print(d)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 4.py
Enter a string:hello
hello    1
```

PROGRAM-4

Prompt the user for a list of integers. For all values greater then 100 store “over” instead.

Program code:

```
x=int(input("enter size"))
li=[]
for I in range(0,x):
el=int(input())
if(el>=100):
li.append("over")
else:
li.append(el)
        print(li)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 100.py
Enter how many numbers
3
Enter the numbers
170
24
60
The Numbers are
['Over', 24, 60]
```

PROGRAM-5

Store the list of first names. Count the occurance of ”a” within the list
Program code:

```
list=["anu","ann","hima"]
count=0
print(list)
foriinlist:
forkini:
if(k=='a'):
count=count+1
        print(count)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 a.py
Enter total number of names: 4
Name:kevin
Name:nikhil
Name:arjun
Name:anurag
'A' occurs  3  times.
```

PROGRAM-6

Enter 2 lists of integers

(a) Whether list are of same length

(b) whether list sums to same value

(c) whether any value occur in both

Source code:

```
l1=[2,4,6,8,10]

l2=[3,5,7,9,10]

print(l1)

print(l2)
```

```
if len(l1)==len(l2):
 print("Lists are of same length")
else:
 print("Lists are of different length")
s1=0
s2=0
for i in range(len(l1)):
 s1=s1+l1[i]
print("Sum of first list is",s1)
for j in range(len(l2)):
 s2=s2+l2[j]
print("Sum of second list is",s2)
if (s1==s2):
 print("Sum of lists is same")
else:
 print("Sum of lists are different")
for i in l1:
 if i in l2:
 print(i,"occurs in both list")
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 7.py
[2, 4, 6, 8, 10]
[3, 5, 7, 9, 10]
Lists are of same length
Sum of first list is 30
Sum of second list is 34
Sum of lists are different
10 occurs in both list
```

PROGRAM-7

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

Program code:

```
s=input("enter a string\n")
print("entered string is:",s)
a=s[0]
str=s.replace(a,"$")
strl=a+str[1:]
print(strl
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 8.py
Enter a string:onion
Original string: onion
String: oni$n
```

PROGRAM-8

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

Program code:

```
a=input("enterastring")
print(a)
a1=a[0]
a2=a[-1]
print(a1)
print(a2)
rev=(a2+a[1:len(a)-1]+a1)
print(rev)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 10.py
nythop
```

PROGRAM-9

Accept the radius from user and find area of circle.
Program code:

```
p=int(input("enter the radius"))
a=3.14*p*p
print(a)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 circle.py
enter radius5
78.5
```

PROGRAM-10

Find biggest of 3 numbers entered.
Program code:

```
a=int(input("enter ist number"))
b=int(input("enter 2nd number"))
c=int(input("enter 3rd number"))
ifa>b and a>c:
print(a)
ifc>b and c>a:
print(c)
else:
        print(b)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 greater.py
enter first number45
enter second number23
enter third number67
the greatest number is 67
```

PROGRAM-11

Accept a filename from user and print extension of that.

Program code:

```
Import os
a=input("Enter the filename\n")
print(os.path.splitext(a))
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 ext.py
enter a file nameext.py
the extension of ext.py is ('ext', '.py')
```

PROGRAM-12

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

Program code:

```
list1=[]
string=input("Enter colors separated by comma:\n")
for I in string.split(","):
list1.append(i)
print("First and last colors in the list are",list1[0],"and",list1[-1])
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 fl.py
first and last colours are:
red & yellow
```

PROGRAM-13

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

Program code:

```
a=int(input("Input an integer:"))
n1=(a*1)
n2=(a*11)
n3=(a*111)
print(n1+n2+n3)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 nnn.py
Input an integer : 4
492
```

PROGRAM-14

Print out all colors from color- list1 not contained in color-list2.

Program code:

```
list1=["red","green","blue","yellow"]
list2=["black","white","cyan","blue","red"]
l3=[]
print(list1)
print(list2)
foriinlist1:
ifinotinlist2:
l3.append(i)
print("Elementspresentinlist1butnotinlist2are")
print(l3)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 list4.py
['red', 'green', 'blue', 'yellow']
['black', 'white', 'cyan', 'blue', 'red']
Elements present in list1 but not in list2 are
['green', 'yellow']
```

PROGRAM-15

Create a single string separated with space from 2 strings swapping the character at position

Program code:

```
str1=input("Enter string1:")
str2=input("Enter string2:")
temp=str1[0]
str1=str1.replace(str1[0],str2[0])
str2=str2.replace(str2[0],temp)
str=str1+" "+str2
print(str)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 16.py
Enter first string:hello
Enter second string:oii
oello  hii
```

PROGRAM-16

Sort a dictionary in ascending and descending order

Program code:

```
d={1:2,3:4,4:3,2:1,0:0}
list1=list(d.items())
dict=dict(list1)
print("Dictionary=",dict)
list1.sort()
print('Ascending order is',list1)
list1=list(d.items())
list1.sort(reverse=True)
print('Descending order is',list1)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 17.py
[('a', 1), ('c', 3), ('d', 2), ('b', 4)]
Ascending order is
 [('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
 [('d', 2), ('c', 3), ('b', 4), ('a', 1)]
```

PROGRAM-17

Merge 2 dictionaries

Program code:

```
d2={"sex":"Female","qualification":"PG"}
d1.update(d2)
        print(d)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 18.py
{'Name': 'Ardhra', 'Age': 25, 'Gender': 'F', 'Qualification': 'PG'}
```

PROGRAM-18

Find gcd of 2 numbers

Program code:

```
a=int(input("enter 1st number"))
b=int(input("enter 2nd number"))
z=min(a,b)
for I in range(1,z+1):
if((a % i== 0)and(b % i==0)):
gcd=i
print("gcd is=",gcd)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 gcd.py
Enter 1st number
30
Enter 2nd number
23
GCD is  1
```

PROGRAM-19

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

Program code:

```
list=[12,13,14,15,16,21]
l1=[]
print(list)
print("New list")
for i in list:
    if i%2!=0:
        l1.append(i)
print(l1)
```

Output:

```
stud@debian:~/dhanik/python/total$ python3 remove.py
removing even number [1, 3, 5]
```

COURSE OUTCOME 2

PROGRAM-20

Program to find the factorial of a number

Program code:

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

Output:

```
enter the number4
24
```

PROGRAM-21

Generate fibonaci series of N terms

Program code:

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

Output:

```
Enter a number:4
0
1
1
2
```

PROGRAM-22

Find the sum of all items in list?

Program code:

```
list=[1,2,3,4,5,6,7,8,9,10]
sum=0
for i in list:
   sum=sum +int(i)
print("sum:" ,sum)
```

Output:

```
15
```

PROGRAM-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 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:

```
68
78
80
92
[4624, 6084, 6400, 8464]
```

PROGRAM-24

Display the given pyramid with step number accepted from user.

Program code:

```
l=int(input('Enter the limit:'))
for i in range(1,l+1):
   for j in range(1,i+1):
      c=i*j
      print(c,end=" ")
   print("\n")
```

Output:

```
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
```

PROGRAM-25

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

Program 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:

```
Enter a string:hey
h       : 1
e       : 1
y       : 1
```

PROGRAM-26

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

Program code:

```
string=input("Enter a string:")
if(string[-3:]=="ing"):
 string+="ly"
else:
 string+="ing"
print(string)
```

Output:

```
Enter a string:find
finding
```

PROGRAM-27

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

Program 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:4
Enter the words:
helloo
hii
hey
hy
Length of longest word is 6
```

PROGRAM-28

Construct following patterns using nested loop

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

Source code:

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

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

Output:

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

PROGRAM-29

Generate all factors of a number.

Program code:

```
n=int(input("Enter a number:"))
print("Factors are")
for i in range(1,n+1):
  if(n%i==0):
    print(i)
```

Output:

```
Enter a number:12
Factors are
1
2
3
4
6
12
```

COURSE OUTCOME 3

PROGRAM-30

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 code:

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

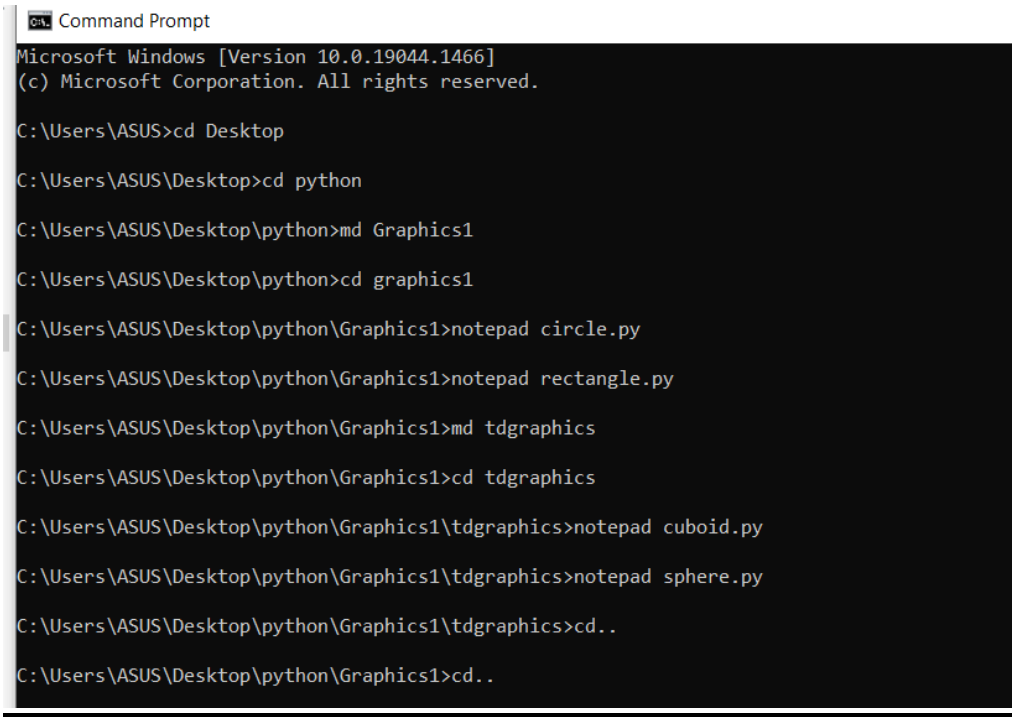
```
def area_cuboid(l,b,h):
 return 2*(l*h + b*h + l*b)
def volume_cuboid(l,b,h):
 return l*b*h
```

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:



```
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

PROGRAM-31

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

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:

```
stud@debian:~/dhanik$ python3 rect1.py
enter length of 1st rectangle3
enter breadth 1st rectangle4
enter length of 2nd rectangle6
enter breadth of 2nd rectangle2
area of 1st rectangle is: 12
perimeter is: 14
area of 2nd rectangle is: 12
Area of second rectangle is greater
stud@debian:~/dhanik$
```

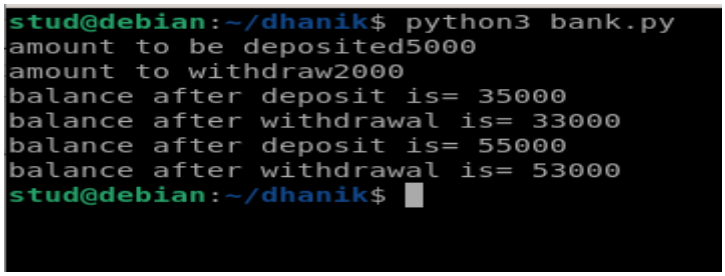
PROGRAM-32

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.

Program 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:



PROGRAM-33

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

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)
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:

```
stud@debian:~/dhanik$ python3 rect1.py
enter length of 1st rectangle3
enter breadth 1st rectangle4
enter length of 2nd rectangle6
enter breadth of 2nd rectangle2
area of 1st rectangle is: 12
perimeter is: 14
area of 2nd rectangle is: 12
Area of second rectangle is greater
stud@debian:~/dhanik$
```

PROGRAM-34

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

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)
```

PROGRAM-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.

Program 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","Percy Jacks","JK Rick",200,100)
p.display3()
```

Output:

```
stud@debian:~/dhanik$ python3 book.py
Percy Jacks
JK Rick
Percy Jacks
JK Rick
200
100
stud@debian:~/dhanik$
```

COURSE OUTCOME 5

PROGRAM-36

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

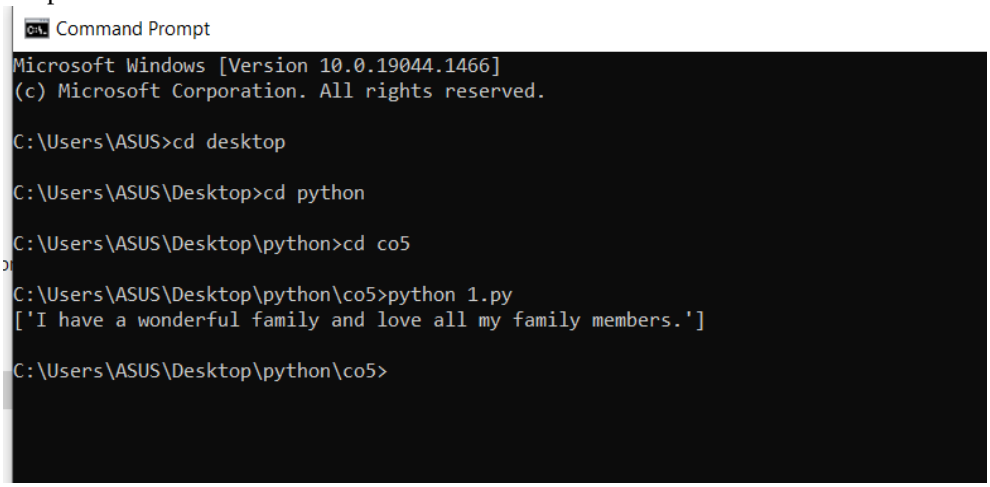
Program code:

```
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:



PROGRAM-37

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

Program 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: