

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577



FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: CHANCHAL K SUNIL

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1

Batch: A

Roll No: 47

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM

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 CHANCHAL K SUNIL 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:
		COURSE OUTCOME 1		
1	28/10/21	Display future leap years from current year to a final year entered by the user.	5	
2	28/10/21	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	6	
3	28/10/21	Count the occurrences of each word in a line of text	8	
4	28/10/21	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	8	
5	10/11/21	Store a list of first names. Count the occurrences of 'a' within the list	9	
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	9	
7	10/11/21	Get a string from an input string where all occurrences of first character replaced with '\$', except first character.	11	
8	10/11/21	Create a string from given string where first and last characters exchanged	12	
9	10/11/21	Accept the radius from user and find area of circle.	12	
10	11/11/21	Find biggest of 3 numbers entered.	13	
11	11/11/21	Accept a file name from user and print extension of that.	13	

12	11/11/21	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	14	
13	11/11/21	Accept an integer n and compute $n+nn+nnn$.	14	
14	11/11/21	Print out all colors from color-list1 not contained in color-list2.	15	
15	17/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	15	
16	17/11/21	Sort dictionary in ascending and descending order.	16	
17	17/11/21	Merge two dictionaries.	16	
18	17/11/21	Find gcd of 2 numbers.	17	
19	17/11/21	From a list of integers, create a list removing even numbers.	17	
		COURSE OUTCOME 2		
1	25/11/12	Program to find the factorial of a number	18	
2	25/11/12	Generate Fibonacci series of N terms	18	
3	25/11/12	Find the sum of all items in a list	19	
4	25/11/12	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	19	
5	02/12/12	Display a pyramid with step number accepted from user.	20	
6	02/12/12	Count the number of characters (character frequency) in a string	21	
7	02/12/12	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	21	

8	09/12/21	Accept a list of words and return length of longest word.	22	
9	09/12/21	Construct a pattern using nested loop	22	
10	09/12/21	Generate all factors of a number.	23	
		COURSE OUTCOME 3		
1	29/01/22	Work with built-in packages	24	
2	29/01/22	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.	24	
		COURSE OUTCOME 4		
1	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.	27	
2	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	28	
3	13/01/22	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	29	
4	20/01/22	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	30	
5	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.	31	

		COURSE OUTCOME 5		
1	03/02/22	Write a Python program to read a file line by line and store it into a list.	32	
2	03/02/22	Write python program to read each row from a given csv file and print a list of strings.	32	

COURSE OUTCOME 1

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

SOURCE CODE

```
print("print the leap years between the two given years")
print("enter the startyear")
startyear= int(input())
print("enter the endyear")
endyear= int(input())
print("list of leap years is") for year in range(startyear,endyear):
    if(0 == year % 4) and (0 != year % 100) or (0 == year % 400):
        print (year)
```

OUTPUT

```
| print the leap years between the two given years
| enter the startyear
| 2020
| enter the endyear
| 2050
| list of leap years is
| 2020
| 2024
| 2028
| 2032
| 2036
| 2040
| 2044
| 2048
| —
```

2. List comprehensions:

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

SOURCE CODE

```
list1=[0,-8,4,3,9,-4,-2,3,7,-4]
list2=[]
for i in list1:
    if i>=0:
        list2.append(i)
print(list2)
```

OUTPUT

```
[0, 4, 3, 9, 3, 7]
```

(b) Square of N numbers.

SOURCE CODE

```
list1=[2,5,8,7]
for i in list1:
    print(i**2)
```

OUTPUT

```
4
25
64
49
```


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

SOURCE CODE

```
s = input("enter any string ")
vowel=[]
for i in s:
    if (i in "AEIOUaeiou"):
        vowel.append(i)
print(vowel)
```

OUTPUT

```
enter any string  Elephant
['E', 'e', 'a']_
```

(d) List ordinal value of each element from a given word.

SOURCE CODE

```
x=input("Enter a word\n")
list1=[]
for i in x:
    y=ord(i)
    list1.append(y)
print(list1)
```

OUTPUT

```
Enter a word
haii
[104, 97, 105, 105]
```

3) Count the occurrence of each word in a line of text.**SOURCE CODE**

```
list1=[ ]
ulist=[ ]
string=input("Enter a line of text :")
for i in string.split(" "):
    list1.append(i)
    if i not in ulist:
        ulist.append(i)

for i in ulist:
    print(i,"\t",list1.count(i))
```

OUTPUT

```
Enter a line of text :Home Sweet Home
Home      2
Sweet     1
```

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

```
list=[]
for i in range(3):
    j=int(input())
    if(j>100):
        list.append("over")
    else:
        list.append(j)
print(list)
```

OUTPUT

```
25
102
87
[25, 'over', 87]
```

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

SOURCE CODE

```
list=[ ]
p=0
for i in range(3):
    j=input()
    list.append(j)
for i in list:
    s=i.count('a')

    p=p+s
print(p)
```

OUTPUT

```
Chanju
Keerthana
Namitha
5
```

6) Enter 2 lists of integers.

SOURCE CODE

```
list1=[1,3,5,7,9]
list2=[1,4,4,6,10]
#checking whether length of the list is same or not
```

```
print("List 1 is")
print(list1)
print("List 2 is")
print(list2)
x=len(list1)
print("length of list1 is")
print(x)
y=len(list2)
print("length of list2 is")
print(y)
if(x==y):
    print("Length of both list is same\n")
else:
    print("Length is not same\n")

#checking whether list sums to the same value
print("sum of elements in list1 is")
s=0
for i in list1:
    s=s+i
print(s)
print("sum of elements in list2 is")
t=0
for j in list2:
    t=t+j
print(t)
if(s==t):
    print("sum of both lists is same")
else:
    print("sum is not same")
#checking whether any value occur in both
flag=0
```

```

for i in list1:
    if (i in list2):
        flag=flag+1
    print("value present in both list is",i)
if(flag==0):
    print("no value is common in the lists")

```

OUTPUT

```

List 1 is
[1, 3, 5, 7, 9]
List 2 is
[1, 4, 4, 6, 10]
length of list1 is
5
length of list2 is
5
Length of both list is same

sum of elements in list1 is
25
sum of elements in list2 is
25
sum of both lists is same
value present in both list is 1

```

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

SOURCE CODE

```

str1=input("enter a string ")
print("original string :",str1)
char=str1[0]
str2=str1.replace(char,'$')
str3=char+str2[1:]
print("Replaced string :",str3)

```

OUTPUT

```
enter a string  onion
original string : onion
Replaced string_ : oni$n
```

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

SOURCE CODE

```
s="python"
x=s[0]
y=s[-1]
n=len(s)
string=y+s[1:n-1]+x
print("origial string: ",s)
print("String after exchanging: ",string)
```

OUTPUT

```
origial string:  python
String after exchanging:  nythop
```

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

SOURCE CODE

```
x=int(input('Enter the radius: '))
A=3.14*x*x
print("Area of the circle is",A)
```

OUTPUT

```
Enter the radius: 5
Area of the circle is 78.5
```

10) Find the biggest of three numbers entered.**SOURCE CODE**

```
x=input('enter three numbers')
y=input()
z=input()
a=int(x)
b=int(y)
c=int(z)
if(a>b):
    if a>c :
        print(a, "is big")
    else:
        print(c," is big")
else:
    if b>c:
        print(b," is big")
    else:
        print(c,"is big")
```

OUTPUT

```
enter three numbers: 2
6
8
8 is big
```

11) Accept a file name from the user and print extension of that.**SOURCE CODE**

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

OUTPUT

```
enter a file name: big.py
The extension of file big.py is ('big', '.py')
```

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

SOURCE CODE

```
s=input("Enter color names separated by commas: ")
list1=s.split(',')
print(list1)
print("First colour is",list1[0])
print("Last colour is",list1[-1])
```

OUTPUT

```
Enter color names separated by commas: Green,Red,Blue,Yellow
['Green', 'Red', 'Blue', 'Yellow']
First colour is Green
Last colour is Yellow
```

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

SOURCE CODE

```
x=int(input("enter a number"))
a=str(x)
b=a+a
c=a+a+a
d=x+int(b)+int(c)
print(d)
```

OUTPUT

```
enter a number: 2
246
```


14) Print out all colors from color list 1 not contained in color list2.

SOURCE CODE

```
list1=["green","white","black","yellow"]
list2=["violet","black","red","rose","yellow"]
list3=[]
for i in list1:
    if(i not in list2):
        list3.append(i)
print("Color list 1",list1)
print("Color list 2",list2)
print("Colors from list 1 not contained in list 2: ",list3)
```

OUTPUT

```
Color list 1 ['green', 'white', 'black', 'yellow']
Color list 2 ['violet', 'black', 'red', 'rose', 'yellow']
Colors from list 1 not contained in list 2: ['green', 'white']
```

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

```
Enter string 1:SWEET
Enter string 2:DREAMS
DWEET SREAMS
```

16) Sort dictionary in ascending and descending order.

SOURCE CODE

```
d = {'a': 30, 'b': 40, 'c': 50, 'd': 60}
print("The original dictionary: ", d)
a = dict(sorted(d.items(), reverse = True, key=lambda x: x[1]))
print("After sorting by value in reverse order: ", a)
a = dict(sorted(d.items(), key=lambda x: x[1]))
print("After sorting by value in ascending order: ", a)
```

OUTPUT

```
The original dictionary: {'a': 30, 'b': 40, 'c': 50, 'd': 60}
After sorting by value in reverse order: {'d': 60, 'c': 50, 'b': 40, 'a': 30}
After sorting by value in ascending order: {'a': 30, 'b': 40, 'c': 50, 'd': 60}
```

17) Merge 2 dictionaries.

SOURCE CODE

```
d1={'a',600,'b',700}
d2={'x',500,'y',400}
d=d1.copy()
d.update(d2)
print("Merged dictionary is ",d)
```

OUTPUT

```
Merged dictionary is {'b', 'y', 'a', 400, 'x', 500, 600, 700}
```

18) Find gcd of 2 numbers.**SOURCE CODE**

```

x=int(input("Enter first digit:"))
y=int(input("Enter second digit:"))
z=min(x,y)
for i in range (1,z+1):
    if(x%i==0 and y%i==0):
        print(i)
gcd=(i)
print("gcd=",gcd)

```

OUTPUT

```

Enter first digit:4
Enter second digit:20
1
2
4
gcd= 4

```

19) From a list of integers create a list removing even numbers.**SOURCE CODE**

```

list1=[2,5,7,8,9,3,15]
list2=[ ]
for i in list1:
    if(i%2!=0):
        list2.append(i)
print("list after removing even integers :",list2)

```

OUTPUT

```

list after removing even integers : [5, 7, 9, 3, 15]

```

COURSE OUTCOME 2**1) Program to find the factorial of a number.****SOURCE CODE**

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

OUTPUT

```
enter a number :5
Factorial is  120
```

2) Generate Fibonacci series of N terms.**SOURCE CODE**

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

OUTPUT

```
Enter a number: 6
0
1
1
2
3
5
```

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

SOURCE CODE

```
list1=[2,8,5,11,4]
s=0
for i in list1:
    s=s+i
print("The sum of all items in a list is",s)
```

OUTPUT

```
The sum of all items in a list is 30
```

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

```
list1=[ ]
for i in range(1000,10000,1):
    for j in range(32,100,1):
        if i==j*j :
            string=str(i)
            if int(string[0])%2==0 and int(string[1])%2==0 and int(string[2])%2==0 and int(string[3])%2==0:
                list1.append(i)
print(list1)
```

OUTPUT

```
[4624, 6084, 6400, 8464]
```

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

SOURCE CODE

```
x=int(input("Enter the number: "))  
for j in range(0,x+1):  
    for i in range(1,j+1):  
        i=j*i;  
        print(i,end=" ");  
    print("\n")
```

OUTPUT

```
Enter the number: 4
```

```
1
```

```
2 4
```

```
3 6 9
```

```
4 8 12 16
```

6) Count the number characters in a string**SOURCE CODE**

```
x=input("Enter a string: ")
c=0
for i in x:
    c=c+1

print("The number of characters in the given string is",c)
```

OUTPUT

```
Enter a string: Chanchal
The number of characters in the given string is 8
```

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

```
x=input("enter the string:")
y=len(x)
if(y>2):
    if(x[-3:]=='ing'):
        print(x+"ly")
    else:
        print(x+"ing")
else:
    print("not sufficient")
```

OUTPUT

```
enter the string:food
fooding

enter the string:wing
wingly
```

8) Accept a list of word and return the length of the largest word.

SOURCE CODE

```
list=[ ]  
length=[ ]  
print("enter 5 words")  
for i in range (5):  
    str=input()  
    list.append(str)  
for j in list:  
    length.append(len(j))  
print("length of longest word is:",max(length))
```

OUTPUT

```
enter 5 words  
cat  
peacock  
butterfly  
rabit  
kitten  
length of longest word is: 9
```

9) Construct following pattern using nested loop.

SOURCE CODE

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


OUTPUT

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

10) Generate all factors of a number.

SOURCE CODE

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

OUTPUT

```
Enter a number: 28  
1  
2  
4  
7  
14  
28
```

COURSE OUTCOME 3**1) Work with built-in packages.****SOURCE CODE**

```
import time
import datetime
today=datetime.date.today()
print(f"The time is {time.ctime()} and date is {today}")
```

OUTPUT

```
The time is Tue Mar  1 05:02:37 2022 and date is 2022-03-01
```

2) 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****Graphics\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("Perimeter 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("Perimeter 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("Permter of a spere with radius 10 is ",sphere.perimeter_sphere(10))

```

OUTPUT

```

C:\Users\chanc\OneDrive\Desktop>cd Python
C:\Users\chanc\OneDrive\Desktop\Python>md Graphics
C:\Users\chanc\OneDrive\Desktop\Python>cd Graphics
C:\Users\chanc\OneDrive\Desktop\Python\Graphics>notepad circle.py
C:\Users\chanc\OneDrive\Desktop\Python\Graphics>notepad rectangle.py
C:\Users\chanc\OneDrive\Desktop\Python\Graphics>md tdgraphics
C:\Users\chanc\OneDrive\Desktop\Python\Graphics>cd tdgraphics
C:\Users\chanc\OneDrive\Desktop\Python\Graphics\tdgraphics>notepad cuboid.py
C:\Users\chanc\OneDrive\Desktop\Python\Graphics\tdgraphics>notepad sphere.py
C:\Users\chanc\OneDrive\Desktop\Python\Graphics\tdgraphics>cd ..
C:\Users\chanc\OneDrive\Desktop\Python\Graphics>cd ..
C:\Users\chanc\OneDrive\Desktop\Python>notepad driver.py
C:\Users\chanc\OneDrive\Desktop\Python>python driver.py
Area of a circle with radius 10 is : 314.1592653589793
Permter of a circle with radius 10 is 62.83185307179586

Area of a Rectangle with length and width 10 is : 100
Permter 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
Permter 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.len=length
        self.br=breadth

    def area(self):
        return self.len*self.br

    def perimeter(self):
        return (self.len+self.br)*2

r1 = Rectangle(10,6)
r2 = Rectangle(5,2)

x=r1.area()
y=r2.area()

print("Area of r1 is",x)
print("Area of r2 is",y)

if(x==y):
    print("Both rectangles have the same area")

else:

    print("Areas are not equal")
```

OUTPUT

```
Area of r1 is 60
Area of r2 is 10
Areas are not equal
```

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,typ_ac,balance):
        self.acc_no=acc_no
        self.name=name
        self.typ_ac=typ_ac
        self.balance=balance

    def withdraw(self,x):
        self.balance=self.balance-x

    def deposit(self,y):
        self.balance=self.balance+y

    def print(self):
        q=self.balance
        print("CUSTOMER WITH A/C No",self.acc_no,"A/C NAME",self.name,"A/C TYPE",self.typ_ac,"HAS BALANCE",q)

ac1=Bank(40112,"chanju","savings",10000)
ac2=Bank(40113,"keerthu","savings",5000)
ac3=Bank(40114,"ankitha","savings",15000)

ac1.withdraw(1000)
ac1.deposit(500)
ac2.deposit(4000)
ac3.withdraw(1000)
ac3.deposit(100)

ac1.print()
ac2.print()
ac3.print()
```

OUTPUT

```
CUSTOMER WITH A/C No 40112 A/C NAME chanju A/C TYPE savings HAS BALANCE 9500
CUSTOMER WITH A/C No 40113 A/C NAME keerthu A/C TYPE savings HAS BALANCE 9000
CUSTOMER WITH A/C No 40114 A/C NAME ankitha A/C TYPE savings HAS BALANCE 14100
```

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.len=length
        self.br=breadth

    def area(self):
        return self.len*self.br

    def perimeter(self):
        return (self.len+self.br)*2

    def __lt__(self,rr):
        if(self.br*self.len<rr.br*rr.len):
            return True
        else:
            return False

r1 = Rectangle(20,10)
r2 = Rectangle(15,2)

x=r1.area()
y=r2.area()

print("Area of r1 is",x)
print("Area of r2 is",y)
if(r1<r2):
    print("r2 has more area")
else:
    print("r1 has more area")
```

OUTPUT

```
Area of r1 is 200
Area of r2 is 30
r1 has more area
```

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, hour, minute, second):
        self.__hr=hour
        self.__min=minute
        self.__sec=second

    def __add__(self, ad):
        x=self.__hr+ad.__hr
        y=self.__min+ad.__min
        z=self.__sec+ad.__sec
        print("t1+t2 is",x,":",y,":",z)

t1=Time(2,10,20)
t2=Time(1,20,5)
t1+t2
```

OUTPUT

```
t1+t2 is 3 : 30 : 25
```


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:
    def __init__(self,name):
        self.name=name
class Book(Publisher):
    def __init__(self,name,author,title):
        super().__init__(name)
        self.title=title
        self.author=author
class Python(Book):
    def __init__(self,name,author,title,price,page):

        super().__init__(name,author,title)
        self.price=price
        self.pages=page
    def Print(self):
        print(P.name)
        print(P.title)
        print(P.author)
        print(P.price)
        print(P.pages)
P=Python('KHANNA Publishers','Taming Python','Jeeva Jose',330,350)
P.Print()
```

OUTPUT

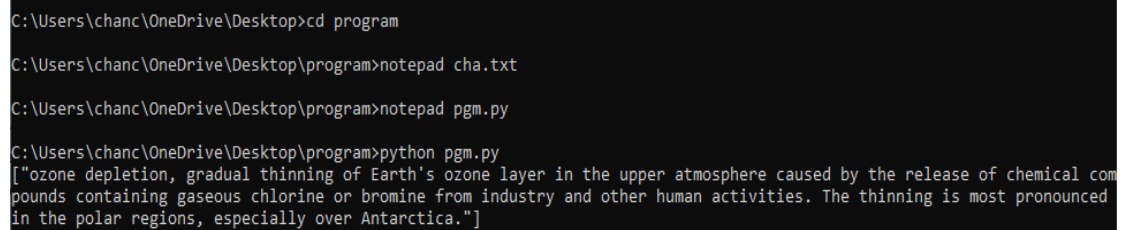
```
KHANNA Publishers
Jeeva Jose
Taming Python
330
350
```

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

OUTPUT


```
C:\Users\chanc\OneDrive\Desktop>cd program
C:\Users\chanc\OneDrive\Desktop\program>notepad cha.txt
C:\Users\chanc\OneDrive\Desktop\program>notepad pgm.py
C:\Users\chanc\OneDrive\Desktop\program>python pgm.py
['ozone depletion, gradual thinning of Earth's ozone layer in the upper atmosphere caused by the release of chemical compounds containing gaseous chlorine or bromine from industry and other human activities. The thinning is most pronounced in the polar regions, especially over Antarctica.']
```

2) Write a python program to read each row from a given CSV file and print a list of strings.

SOURCE CODE

final.csv

Name,Place,DOB

Chanchal,Kannur,06/10/2000

Keerthana,Wayanad,12/05/2000

Ankitha,Malappuram,04/03/2000

prog.py

```
import csv

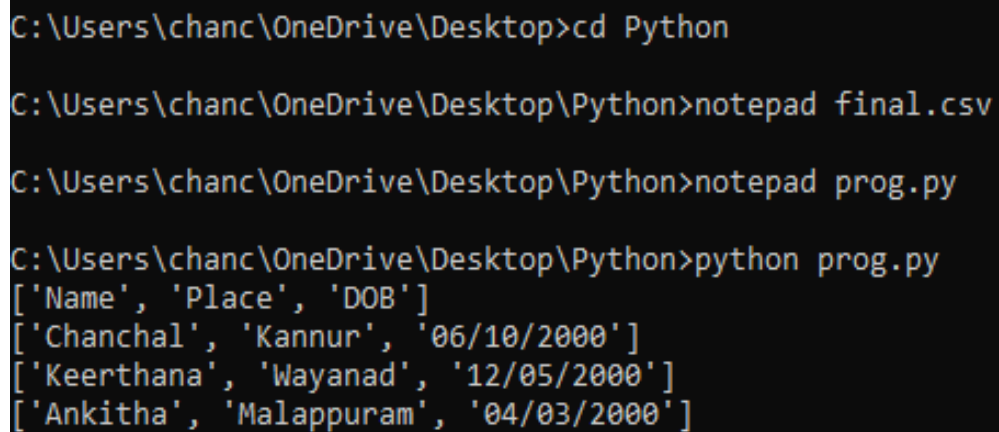
with open('final.csv', 'r') as file:

    reader = csv.reader(file)

    for row in reader:

        print(row)
```

OUTPUT



The screenshot shows a Windows command prompt with the following commands and output:

```
C:\Users\chanc\OneDrive\Desktop>cd Python
C:\Users\chanc\OneDrive\Desktop\Python>notepad final.csv
C:\Users\chanc\OneDrive\Desktop\Python>notepad prog.py
C:\Users\chanc\OneDrive\Desktop\Python>python prog.py
['Name', 'Place', 'DOB']
['Chanchal', 'Kannur', '06/10/2000']
['Keerthana', 'Wayanad', '12/05/2000']
['Ankitha', 'Malappuram', '04/03/2000']
```