FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

LABORATORY RECORD

20MCA131 - PROGRAMMING LAB

Name: **ANIL KURIAN**

Branch: **MASTER OF COMPUTER APPLICATION**

Semester: **1** **Batch: 2021 A** **Roll No: 22**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name : ANIL KURIAN

Branch : MASTER OF COMPUTER APPLICATIONS

Semester : 1 **Roll No: 22**

University Exam.Reg. No: FIT21MCA-2022

CERTIFICATE

Certified that this is the Bonafide record of the Practical work done by Mr. **ANIL KURIAN(FIT21MCA-2022)** in the **20MCA131-PROGRAMMING** Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.

Signature of Staff in Charge

Name:

Date:

Signature of H.O.D

Name:

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

CONTENT

12	**11/11/2021**	Create a list of colors from comma-separated color names entered by user. Display first and last colors.		
13	**11/11/2021**	Accept an integer n and compute n+nn+nnn.		
14	**17/11/2021**	Print out all colors from color-list1 not contained in color-list2.		
15	**17/11/2021**	Create a single string separated with space from two strings by swapping the character at position 1.		
16	**17/11/2021**	Merge two dictionaries.		
17	**17/11/2021**	Find gcd of 2 numbers.		
18	**17/11/2021**	From a list of integers, create a list removing even numbers.		
CO2				
19	**25/11/2021**	Program to find the factorial of a number		
20	**25/11/2021**	Generate Fibonacci series of N terms		
21	**25/11/2021**	Find the sum of all items in a list		
22	**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.		
23	**02/12/2021**	Display the given pyramid with step number accepted from user.		
24	**02/12/2021**	Count the number of characters (character frequency) in a string.		
25	**02/12/2021**	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.		
26	**09/12/2021**	Accept a list of words and return length of longest word.		

27	**09/12/2021**	Construct following pattern using nested loop. *		
28	**09/12/2021**	Generate all factors of a number.		
		CO3		
29	**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)		
		CO4		
30	**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.		
31	**29/01/2022**	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.		
32	**29/01/2022**	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.		
33	**20/01/2022**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time		

34	**29/01/2022**	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.		
CO5				
35	**03/02/2022**	Write a Python program to read a file line by line and store it into a list.		
36	**03/02/2022**	Write a Python program to read each row from a given csv file and print a list of strings.		

COURSE OUTCOME 1

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

Source code

```
print("Enter leap year
between given two 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

```
ccf@FISATPC0360:~/anil/python$ python3 prgtwo.py
Enter leap year between given two years
Enter end year 2060
List of leap years
2024
2028
2032
2036
2040
2044
2048
2052
2056
ccf@FISATPC0360:~/anil/python$
```

2) List comprehensions:

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

Source code

```
list=[-1,1,7,25,-34,38]
print("Elements in the list are:",list) print("Positive numbers in the list")
for num in list:
    if num>=0:
        print(num)
```

Output

```
ccf@FISATPC0360:~/anil/python$ python3 prgthreea.py
1  7  25  38  ccf@FISATPC0360:~/anil/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

```
ccf@FISATPC0360:~/anil/python$ python3 prgthreeb.py
enter range 5
1
4
9
16
25
ccf@FISATPC0360:~/anil/python$
```

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

Source code

```
s="anil "
list=[]
for i in s:
   if i in "aeiouAEIOU":
        list.append(i)
print("vowels in the list are:")
print(list)
```

Output

```
ccf@FISATPC0360:~/anil/python$ python3 prgthreec.py
['a', 'i']
ccf@FISATPC0360:~/anil/python$ s
```

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

Source code

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

Output

```
ccf@FISATPC0360:~/anil/python$ python3 prgthreed.py
Enter a name:fisat
The ASCII value of the letters in the word is
102
105
115
97
116
ccf@FISATPC0360:~/anil/python$
```

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

Source code

```
list1=[]
list2=[]
x=input("Enter a string:")
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:~/Anil$ python3 prg4.py
Enter a string: home is sweet
          1
home      1
is        1
sweet     1
stud@debian:~/Anil$
```

4) Prompt the user for a list of integers. For all values greater than 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:~/Anil$ python3 prg5.py
Enter an integer: 20
Enter an integer: 66
Enter an integer: 99
Enter an integer: 101
[20, 66, 99, 'Over']
Enter an integer:
```

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

Source code

```
list=['anil','amal','nill'] 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

```
ccf@FISATPC0360:~/anil/python$ python3 prgsix.py
The occurences of 'a' within the list is 4
ccf@FISATPC0360:~/anil/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 in both.

Source code

```
l1=[1,2,3,4]
l2=[5,8,7]
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:~/Anil/Python$ python3 sevpy.py
[1, 2, 3, 4]
[5, 8, 7]
not same length
the sum of the first list is: 10
the sum of the second list is: 20
There is no element in common
stud@debian:~/Anil/Python$
```

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

Source code

```
str=input("Enter a string: ")
print("Original string: ",str)
char=str[0]
str=str.replace(char,'$')
str=char+str[1:]
print("String: ",str)
```

Output

```
stud@debian:~/Anil$ python3 prg.py
enter a stringonion
original string onion
string: oni$n
```

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

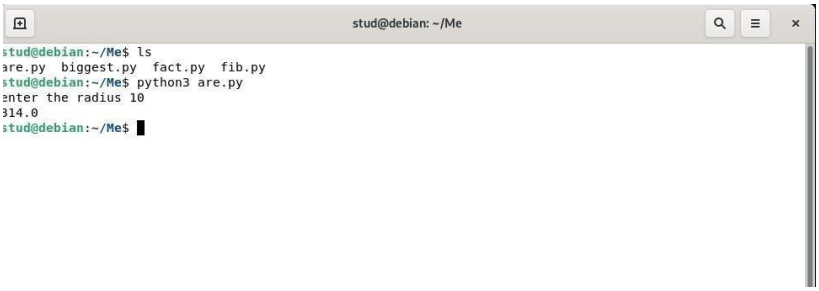
```
stud@debian:~/Anil$ gedit prg9.py
stud@debian:~/Anil$ python3 prg9.py
nythop
```

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

Source code

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

Output



10) Find the biggest of 3 numbers

Source code

```
a=int(input('Enter the value of a:'))
b=int(input('Enter the value of b:'))
c=int(input('Enter the value c:'))
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

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

Source code

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

Output

```
stud@debian:~/Anil$ python3 prg12.py
enter the filename : python.py
The extension of file python.py is ('python', '.py')
stud@debian:~/Anil$
```

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:~/Anil/Python$ python3 thirtepy.py
Ente the size:4
Enter Your Choice:Red
Enter Your Choice:Green
Enter Your Choice:Blue
Enter Your Choice:White
Red
White
stud@debian:~/Anil/Python$
```

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

Source code

```
n=int(input("Enter a 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:~/Anil$ python3 prgint14.py
Enter a number:4
492
```

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

Source code

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

Output

```
stud@debian:~/Anil/Python$ python3 fiftepy.py
red
green
stud@debian:~/Anil/Python$
```

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

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:~/Anil$ python3 16.py
enter 1st string anil
enter 2nd string kiran
 a k
stud@debian:~/Anil$
```

16) Merge twodictionaries.

Source code

```
D1={"Name":"Anil","Age":"21"}
print("Directory 1",D1)
D2={"Gender":"male","Qualification":"BCA"}
print("Directory 2",D2)
D1.update(D2)
print("After merging...")
print(D1)
```

Output

```
stud@debian:~/Anil$ python3 co1prg19.py
Directory 1 {'Name': 'Anil', 'Age': '21'}
Directory 2 {'Gender': 'male', 'Qualification': 'BCA'}
After merging...
{'Name': 'Anil', 'Age': '21', 'Gender': 'male', 'Qualification': 'BCA'}
stud@debian:~/Anil$
```

17) 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:~/Anil$ python3 prg19.py
enter the first number24
enter the second number36
the hcf is 12
stud@debian:~/Anil$
```

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

Source code

```
l1=[1,2,3,4,5,6,7,8,9,10]
print(l1)
l2=[]
for i in range(len(l1)):
 if l1[i]%2!=0:
        l2.append(l1[i])
print("List after removing even elements")
print(l2)
```

Output

```
stud@debian:~/Anil/Python$ python3 twentypy.py
[1, 3, 5, 7]
stud@debian:~/Anil/Python$
```

COURSE OUTCOME 2

19) Program to find the factorial of a number.

Source code

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

Output

20) Generate fibonacci series of N terms.

Source code

```
n=int(input('enter the value:'))
a=0
b=1
print(a)
print(b)
for i in range (2,n):
        c=a+b
        print(c)
        a=b
        b=c
```

Output

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

Source code

```
list=[1,6,3,4,1]
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:~/Anil$ python3 co2prg3.py
15
stud@debian:~/Anil$
```

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

Source code

```
limit1=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:~/Anil$ python3 co2prg4.py
68
78
80
92
[4624, 6084, 6400, 8464]
stud@debian:~/Anil$
```

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

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:~/Anil$ gedit co2prg5.py
stud@debian:~/Anil$ python3 co2prg5.py
Enter a number:4
1

2       4

3       6       9

4       8       12      16

stud@debian:~/Anil$
```

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

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:~/Anil$ python3 co2prg6.py
Enter a string:armstrong
a       : 1
r       : 2
m       : 1
s       : 1
t       : 1
o       : 1
n       : 1
g       : 1
stud@debian:~/Anil$
```

25) 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:~/Anil$ gedit co2prg7.py
stud@debian:~/Anil$ python3 co2prg7.py
Enter a string:subsrtact
subsrtacting
stud@debian:~/Anil$
```

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

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:~/Anil$ gedit co2prg8.py
stud@debian:~/Anil$ python3 co2prg8.py
Enter the range:4
Enter the words:
ok
good
verygood
morning
Length of longest word is 8
stud@debian:~/Anil$
```

27) Construct following pattern using nested loop.

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

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:~/Anil$ gedit co2prg9.py
stud@debian:~/Anil$ python3 co2prg9.py
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

28) Generate all factors of a number.

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:~/Anil$ gedit co2prg10.py
stud@debian:~/Anil$ python3 co2prg10.py
Enter a number:12
Factors are
1
2
3
4
6
12
stud@debian:~/Anil$
```

COURSE OUTCOME 3

29) 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


Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\owner\Desktop\LAB MCA\PYTHON\record\co3>2.py
Area of rectangle :144
Area of circle :36
Area of sphere :1808.6399999999999
Area of cuboid :95551488
Perimeter of rectangle :48
Perimeter of circle :37.68
Diameter of sphere :24
Periameter of cuboid :184
C:\Users\owner\Desktop\LAB MCA\PYTHON\record\co3>

COURSE OUTCOME 4

30) 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

```
Enter length of rectangle1: 4
Enter breadth of rectangle1: 6
Area: 24
Perimeter: 20
Enter length of rectangle2: 2
Enter breadth of rectangle2: 3
Area: 6
Perimeter: 10
First rectangle is larger
```

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

Source code

```
class bank:
def_init_(self,acc_no,name,acc_type,bal):
        self.acc_no=acc_no
        self.name=name
        self.acc_type=acc_type
        self.bal=bal


  def deposit(self):
        self.bal=self.bal+y
        return self.bal
  def withdraw(self):
        return self.bal-y
  def display_balance(self):
        return self.bal
acc1=bank("b11","Ann","Savings",50000)
while(1):
  print("1.Deposit\n2.Withdraw\n3.Display balance\n4.Exit\n")
  ch=int(input("Enter your choice:"))
  if ch==1:
        amt=int(input("Enter the amount:"))
        b=acc1.deposit(amt)
        print("Current balance:",b)
```

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

Output

```
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:1000
Current balance: 51000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:4
```

32) 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

```
Enter length of rectangle1: 1
Enter breadth of rectangle1: 3
Enter length of rectangle2: 5
Enter breadth of rectangle2: 8
Second rectangle is larger
```

33) 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
```

34) 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","Java programming","E
Balagurusami",500,1200)
q.display3()
```

Output

```
Taming Python
jeeva jose
100
500
Java programming
E Balagurusami
500
1200
```

COURSE OUTCOME 5

35) 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

```
["Kerala, a state on India's tropical Malabar Coast, has nearly 600km of Arabian
 Sea shoreline. It's known for its palm-lined beaches and backwaters, a network
of canals. Inland are the Western Ghats, mountains whose slopes support tea, cof
fee and spice plantations as well as wildlife."]
```

36) 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

```
['Name', 'Age', 'Profession']
['John', '30', 'Manager']
['Jerin', '20', 'Accountant']
['Ann', '22', 'Professor']
['Angel', '24', 'Engineer']
['Sree lakshmi', '28', 'Doctor']
```