

# **FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577**



**FOCUS ON EXCELLENCE**

## **20MCA131 PROGRAMMING LAB LABORATORY RECORD**

**Name: ANUPAMA R**

**Branch: MASTER OF COMPUTER APPLICATIONS**

**Semester: 1**

**Batch: A**

**Roll No: 32**

**MARCH 2022**

# **FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)<sup>TM</sup>**

**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 ANUPAMA R(FIT21MCA-2032) 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**

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
<b>CO1</b>				
1	28-10-2021	Display future leap years from current year to a final year entered by user.	1	
2	28-10-2021	List comprehensions: Generate positive list of numbers from a given list of integers Square of N numbers Form a list of vowels selected from a given word List ordinal value of each element of a word (Hint: use ord() to get ordinal values)	1	
3	28-10-2021	Count the occurrences of each word in a line of text.	3	
4	28-10-2021	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.	4	
5	10-11-2021	Store a list of first names. Count the occurrences of 'a' within the list	4	
6	10-11-2021	Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both	5	
7	10-11-2021	Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion ->oni\$n]	6	
8	10-11-2021	Create a string from given string where first and last characters exchanged. [eg: python ->nythop]	6	
9	10-11-2021	Accept the radius from user and find area of circle.	7	
10	11-11-2021	Find biggest of 3 numbers entered.	7	
11	11-11-2021	Accept a file name from user and print extension of that.	8	

<b>Sl No</b>	<b>Date of Experiment</b>	<b>Title of the Experiment</b>	<b>Page No:</b>	<b>Signature of Staff -In - Charge</b>
12	11-11-2021	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	8	
13	11-11-2021	Accept an integer n and compute $n+nn+nnn$ .	9	
14	11-11-2021	Print out all colors from color-list1 not contained in color-list2.	9	
15	17-11-2021	Create a single string separated with space from two strings by swapping the character at position 1.	10	
16	17-11-2021	Sort dictionary in ascending and descending order.	10	
17	17-11-2021	Merge two dictionaries.	11	
18	17-11-2021	Find gcd of 2 numbers.	11	
19	17-11-2021	From a list of integers, create a list removing even numbers.	12	
<b>CO2</b>				
20	25-11-2021	Program to find the factorial of a number.	13	
21	25-11-2021	Generate Fibonacci series of N terms.	13	
22	25-11-2021	Find the sum of all items in a list	14	
23	25-11-2021	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	14	
24	02-12-2021	Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 4 8 12 16	15	
25	02-12-2021	Count the number of characters (character frequency) in a string.	16	
26	02-12-2021	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	17	

<b>Sl No</b>	<b>Date of Experiment</b>	<b>Title of the Experiment</b>	<b>Page No:</b>	<b>Signature of Staff –In – Charge</b>
27	09-12-2021	Accept a list of words and return length of longest word.	17	
28	09-12-2021	Construct following pattern using nested loop * * * * * * * * * * * * * * * * * * * * * * * * *	18	
29	09-12-2021	Generate all factors of a number.	19	
<b>CO3</b>				
30	29-01-2022	Work with built-in packages.	21	
31	29-01-2022	Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)	21	
<b>CO4</b>				
32	13-01-2022	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	24	
33	13-01-2022	Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.	25	

34	13-01-2022	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	26	
35	20-01-2022	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	27	
36	20-01-2022	Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.	28	
<b>CO5</b>				
37	03-02-2022	Write a Python program to read a file line by line and store it into a list.	30	
38	03-02-2022	Write a Python program to read each row from a given csv file and print a list of string.	30	

## **COURSE OUTCOME 1**

- 1) Display future leap years from current year to a final year entered by**

**User.**

**Source code**

```
currentyear=int(input("Enter the current year:"))
finalyear=int(input("Enter the final year:"))
for year in range(currentyear,finalyear):
    if(year%400==0)or(year%100!=0)and(year%4==0):
        print(year)
```

**Output**

```
stud@debian:~/anupamar$ python3 leap.py
enter the starting year2000
enter the ending year2020
leap year
2000
2004
2008
2012
2016
```

- 2) List comprehensions:**

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

**Source code**

```
list1=[-3,3,-12,10,28,0]
for num in list1:
    if num>=0:
        print(num)
```

**Output**

```
stud@debian:~/anupamar$ python3 positive.py
[-2, 3, -12, 10, 28, 0]
positive integers
3
10
28
0
```

**b. Square of N numbers**

**Source code**

```
print("Elements in the list are")
list1=[2,3,4,22,8,5]
for s in list1:
    num=s*s
    print(num)
```

```
stud@debian:~/Downloads$ python3 square.py
[2, 3, 4, 22, 8, 5]
squares
4
9
16
484
64
25
```

**Output**

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

**Source code**

```
L=[]
s=("apple")
for i in s:
    if i in "AaEeIiOoUu":
        L.append(i)
print(L)
```

### Output

```
stud@debian:~/anupamar$ python3 vowels.py
apple
['a', 'e'] -
```

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

### Source code

```
s=("apple")
for i in s:
    print(ord(i))
```

### Output

```
stud@debian:~/Downloads$ python3 ordinal.py
97
112
112
108
101
```

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

### Source code

```
s=input("Enter a string:")
count=dict()
word=s.split()
for i in word:
    if i in count:
        count[i]+=1
    else:
        count[i]=1
print(count)
```

### Output

```
C:\Users\hp\Desktop\python1>python 1co4.py
{'Hello': 1, 'all': 3, 'python': 1, 'is': 2}
```

- 4) Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

### Source code

```
list=[]
n=int(input("enter the limit"))
print("Enter the integer numbers:")
for i in range(0,n):
    j=int(input())
    if j>=100:
        list.append('over')
    else:
        list.append(j)
print(list)
```

### Output

```
stud@debian:~/Downloads$ python3 over.py
enter the limit4
enter the integers:
100
502
20
40
[100, 'over', 20, 40]
```

- 5) Store a list of first names. Count the occurrences of ‘a’ within the list.

### Source code

```
l=[]
r=0
s=("anu,abhi,apple,orange")
for i in s:
    if i in ("aA"):
```

```
r=r+1  
l.append(i)  
print(r)
```

### Output

```
stud@debian:~/Downloads$ python3 count.py  
anu,abhi,apple,orange  
4
```

### 6) Enter 2 lists of integers.Check

- a. whether list are of same length
- b. whether list sums of same value
- c. whether any value occur in both.

### Source code

```
l1=[10,22,4,]  
l2=[12,11,2]  
print(l1)  
print(l2)  
x=len(l1)  
y=len(l2)  
if x==y:  
    print("list are of same length")  
else:  
    print("list are of different length")  
s1=0  
s2=0  
for i in range(x):  
    s1=s1+l1[i]  
print("the sum of 1st list:",s1)  
for j in range(y):  
    s2=s2+l2[j]  
print("The sum of second list:",s2)  
if s1==s2:  
    print("sum of list are same")  
else:  
    print("sum of list are different")  
flag==0;  
for i in l1:
```

```

if I in l2:
    print("common elements:")i
    flag=1
if flag==0:
    print("no common elements")

```

## Output

```
stud@debian:~/Downloads$ python3 length.py
list are of same length
the sum of first list 36
the sum of second list 25
the sum are different
no common in both:
```

- 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

```
str1=input("Enter the string:")
print("Original string:",str1)
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print("Replaced String:",str1)
```

## Output

```
stud@debian:~/Downloads$ python3 replace.py  
enter a string:onion  
original string: onion  
replaced string: oni$n
```

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

## Source code

```
S='python'
t=s[0]
t1=s[-1]
n=len(s)
ns=t1+s[1:n-1]+t
print("Replaced string:",ns);
```

**Output**

```
stud@debian:~/anupamar$ python3 exchange.py
enter a string:python
new string: nythop
```

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

**Source code**

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

**Output**

```
new directory: my dir>
stud@debian:~/Downloads$ python3 area.py
enter the radius2
12.56
```

- 10) Find the biggest of 3 numbers

**Source code**

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

```
stud@debian:~/Downloads$ python3 biggest.py  
enter a10  
enter b15  
enter c7  
15
```

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

### Source code

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

### Output

```
stud@debian:~/anupamar$ python3 extension.py  
enter the file name:pyramid.py  
The extension of file pyramid.py is ('pyramid', '.py')
```

- 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 the color :"))  
for i in str.split(','):  
    colors.append(i)  
print(colors)  
print("first color:",colors[0],"Last color:",colors[-1])
```

### Output

```
stud@debian:~/Downloads$ python3 color.py
enter color nameblue,red,ash,white
blue white
```

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

#### Source code

```
a=(input("enter a number"))

print(" ",a)

b=a+a

print("",b)

c=a+a+a

print(c)

d=int(a)+int(b)+int(c)

print(d)
```

#### Output:

```
stud@debian:~/Downloads$ python3 n+n.py
enter a number:2
2
22
222
246
```

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

#### Source code

```
list1=['red','yellow','blue','green']
list2=['black','red','blue','white']
list3=[]
for i in list1:
```

```
if i not in list2:  
    list3.append(i)  
print(list3)
```

### Output

```
stud@debian:~/anupamar$ python3 listcolour.py  
['red', 'yellow', 'blue', 'green']  
['violet', 'red', 'blue', 'white']  
yellow  
green
```

- 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:")  
a1=b[0]+a[1:]  
b1=a[0]+b[1:]  
c=a1+''+b1  
print(c)
```

### output

```
stud@debian:~/anupamar$ python3 16.py  
Enter string 1:helloo  
Enter string 2:welcome  
welloo helcome
```

- 16) Sort dictionary in ascending and descending order.

### Source code

```
dict1={"a":1,"c":3,"d":2,"b":4}  
l=list(dict1.items())
```

```
print(l)
l.sort()
print("Ascending Order is \n",l)
l=list(dict1.items())
l.sort(reverse=True)
print("Descending order is \n",l)
```

**output**

```
[(d', 2), (c', 3), (a', 1), (b', 4)]
Ascending Order is
[(a', 1), (b', 4), (c', 3), (d', 2)]
Descending order is
[(d', 2), (c', 3), (b', 4), (a', 1)]
```

**17) Merge two dictionaries.**

**Source code**

```
d1={"name":"anu","age":"40"}
d2={"sex":"f","qual":"pg"}
d1.update(d2)
print(d1)
```

**Output**

```
stud@debian:~/Downloads$ python3 dic.py
{'name': 'anu', 'age': '40', 'sex': 'f', 'qual': 'pg'}
```

**18) Find gcd of 2 numbers**

**Source code**

```
x=int(input("Enter the first number:"));
y=int(input("Enter the second number:"));
```

```
if x>y:  
    smallest=y  
else:  
    smallest=x  
for i in range(1,smallest+1):  
    if((x%i==0) and (y%i==0)):  
        gcd=i  
print("gcd is:",gcd)
```

### Output

```
stud@debian:~/anupamar$ python3 gcd.py  
first15  
second5  
gcd( 15 , 5 ): 5
```

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

### Source code

```
list=[12,32,33,45,65,66]  
  
l1=[]  
  
print(list)  
  
print("New list")  
  
for i in list:  
  
    if i%2!=0:  
  
        l1.append(i)  
  
print(l1)
```

### Output

```
stud@debian:~/anupamar$ python3 remove.py  
[12, 32, 33, 45, 65, 66]  
new list  
[33, 45, 65]
```

## **COURSE OUTCOME 2**

- 1) Program to find the factorial of a number.**

**Source code**

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

**Output**

```
stud@debian:~/anupamar$ python3 fact.py  
enter a value for n5  
120
```

- 2) Generate fibonacci series of N terms.**

**Source code**

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

**Output**

```
stud@debian:~/anupamar$ python3 fibo.py
enter the limit6
0
1
1
2
3
5
.....
```

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

**Source code**

```
list=[2,4,3,5]
print("list=",list)
sum=0
for i in list:
    sum=sum+i
print("sum=",sum)
```

**Output**

```
stud@debian:~/anupamar$ python3 sumco2.py
list [2, 4, 3, 5]
sum: 14
```

- 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**

```
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:~/python3$ python3 c04.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

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

Eg n=4

```
1
2 4
3 6 9
4 8 12 16
```

**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:~/anupamar$ python3 pattern.py
enter the limit:5
1

2 4

3 6 9

4 8 12 16

5 10 15 20 25
```

**6) 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:~/anupamar$ python3 c06.py
Enter a string:helloworld
h      : 1
e      : 1
l      : 3
o      : 3
      : 1
w      : 1
r      : 1
d      : 1
```

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

## Source code

```
str1=input("enter a string:")
if str1[-3:]==ing':
    str1=str1+ly'
else:
    str1=str1+'ing'
print("changed string:",str1)
```

## Output

```
stud@debian:~/anupamar$ python3 c07.py
enter a string:character abd string
changed string: character abd stringly
```

**8) Accept a list of words and return length of longest 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

```
stud@debian:~/anupamar$ python3 c08.py
enter 5 words
hello
welcome
python
happy
language
length of longest word is: 8
```

### 9) Construct following pattern using nested loop.

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

### Source code

```
for j in range(0,5):
    for i in range(j+1):
        print('*',end=" ");
    print("\n");
for j in range(j+1,0,-1):
```

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

## output

### **10) 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:~/anupamar$ python3 c10.py
Enter a number:10
Factors are
1
2
5
10
```

### **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 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**

##### **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 sphere with radius 10 is : ",sphere.area_sphere(10))
print("Perimeter of a sphere with radius 10 is ",sphere.perimeter_sphere(10))
```

**Output**

```
C:\Users\hp\Desktop\python1>md graphics
C:\Users\hp\Desktop\python1>cd graphics
C:\Users\hp\Desktop\python1\graphics>notepad circle.py
C:\Users\hp\Desktop\python1\graphics>notepad rectangle.py
C:\Users\hp\Desktop\python1\graphics>md tdgraphics
C:\Users\hp\Desktop\python1\graphics>cd tdgraphics
C:\Users\hp\Desktop\python1\graphics\tdgraphics>notepad cuboid.py
C:\Users\hp\Desktop\python1\graphics\tdgraphics>notepad sphere.py
```

```
C:\Users\hp\Desktop\python1>python driver.py
Area of a circle with radius 10 is :  314.1592653589793
Perimeter of a circle with radius 10 is  62.83185307179586

Area of a Rectangle with length and width 10 is :  100
Perimeter 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
Perimeter 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,l,b):
        self.length=l
        self.breadth=b
    def area(self):
        return self.length*self.breadth
    def peri(self):
        return 2*(self.length+self.breadth)
r1=Rectangle(10,2)
r2=Rectangle(5,9)
x=r1.area()
y=r2.area()
m=r1.peri()
n=r2.peri()
print("area of first rectangle=",x)
print("area of second rectangle=",y)
print("perimeter of first rectangle=",m)
print("perimeter of second rectangle=",n)
if(x<y):
    print("r1 is smaller")
else:
    print("r2 is smaller")
```

## Output

```
C:\Users\hp\Desktop\python1>python 4co1.py
Area of first rectangle 120
Area of second rectangle 80
Perimeter of first rectangle: 52
Perimeter of second rectangle: 36
Area of first rectangle 120 is largest
```

- 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,account_no,name,type_of_account,balance):
        self.acno=account_no
        self.name=name
        self.toa=type_of_account
        self.b=balance
    def withdraw(self,x):
        self.b=self.b-x
    def deposit(self,y):
        self.b=self.b+y
    def print(self):
        print("account number=",self.acno,"name=",self.name,"type of
account=",self.toa,"balance=",self.b)
acc1=bank(123,"anna","fixed",3800)
acc2=bank(124,"ammu","sb",1000)
acc1.withdraw(1000)
acc2.deposit(20000)
acc1.print()
```

```
acc2.print()
```

### Output

```
C:\Users\hp\Desktop\python1>python 4co2.py
.....
Account no: 123
Name: Athira
Account type: Savings
Balance amount: 10000
Balance after deposite: 15000
After withdraw: 14500
.....
Account no: 456
Name: Ryan
Account type: Savings
Balance amount: 20000
Balance after deposite: 22000
After withdraw: 21000
```

- 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.__length = length
        self.__breadth = breadth
    def __lt__(self,rect2):
        if self.__length*self.__breadth < rect2.__length*rect2.__breadth:
            return True
        else:
            return False
r1=Rectangle(5,20)
r2=Rectangle(6,12)
if(r1<r2):
    print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
```

**output**

```
C:\Users\hp\Desktop\python1>python 4co3.py
Area of first rectangle is greater
```

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

**Output**

```
C:\Users\hp\Desktop\python1>python 4co4.py
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(object):
    def __init__(self,name):
        self.name=name
    def display1(self):
        print(self.name)
        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.name)
        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**

```
C:\Users\hp\Desktop\python1>python 4co5.py
ABC Publications
Python
Balaguru Swami
200
800
```

## **COURSE OUTCOME 5**

- 1) Write a Python program to read a file line by line and store it into a list.

### **Source code**

#### **text\_file.txt**

cats ,also called domestic cats are small,carnivorous mammals,of the family felidae,domestic cat are often called house cat when kept as indoor pets,cats have been domesticated for nearly 10,000 years.they are the most popular pets in the world.

#### **5c1.py**

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

### **output**

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn1.py
[ "Cats, also called domestic cats are small, carnivorous mammals, of the family Felidae.', "Domestic cat
s are often called 'house cats' when kept as indoor pets.", 'Cats have been domesticated for nearly 10,00
0 years.', 'They are one of the most popular pets in the world.' ]
PS C:\Users\HP\OneDrive\Desktop\python\co5> []
```

- 2) Write a Python program to read each row from a given csv file and print a list of strings.

### **Source code**

**c05.csv**

```
name place
anu thrissur
ammu palakad
sri kannur
```

**co51.py**

```
import csv
with open('c05.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

**Output**

```
stud@debian:~/anumcaa/python$ python3 pgm40.py
['name place age']
['anu mattom 22']
['alan alkode 21']
['ansa kochi 22']
['ammu kechery 22']
```

