FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ANAGHA VINAYAKAN

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 18**

Register No : FIT21MCA-2018

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done by ***ANAGHA VINAYAKAN(FIT21MCA-2018)*** *in the* ***20MCA131 PROGRAMMING LAB*** *Laboratory towards the partial fulfillment for the award of the Master Of Computer Applications during the academic year2021-2022.*

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
8	09/12/2021	Accept a list of words and return length of longest word.	24	
9	09/12/2021	Construct following pattern using nested loop *	25	
10	09/12/2021	Generate all factors of a number.		
		COURSE OUTCOME 3		
1	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)	28	
		COURSE OUTCOME 4		
1	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.	32	
2	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.	34	

COURSE OUTCOME 1

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

Input

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

Output

```
stud@debian:~/anagha18$ python3 leapyr.py
Enter start year
2002
Enter last year
2021
List of leap years:
2004
2008
2012
2016
2020
```

2.List comprehensions:

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

Input

```
list=[10,-5,4,-8,35,67,-22]
for num in list:
    if num>0:
  print(num)
```

Output

```
stud@debian:~/anagha18$ python3 3a1.py
10
4
35
67
```

(b)Square of N numbers

Input

```
lst=[]
n=int(input("Enter a number:"))
for num in range(1,n+1):
num=num*num
lst.append(num)
 print(lst)
```

Output

```
stud@debian:~/anagha18$ python3 3b.py
Enter a number:4
[1, 4, 9, 16]
```

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

Input

```
L=[]
s="India is my country"
for i in s:
  if i in ("aeiouAEIOU"):
L.append(i)
 print(L)
```

Output

```
stud@debian:~/anagha18$ python3 3c.py
['I', 'i', 'a', 'i', 'o', 'u']
```

(d)List ordinal value of each element of a word

Input

```
ordinal=input("Enter a word:")
print("The ASCII value of the letters in the word is")
for letter in ordinal:
n=ord(letter)
 print(n)
```

output

```
stud@debian:~/anagha18$ python3 3d.py
Enter a word:anagha
The ASCII value of the letters in the word is
97
110
97
103
104
97
```

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

Input

```
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:~/anagha18$ python3 41.py
Enter a string: ammu appu unni appu
        1
ammu    1
appu    2
unni    1
```

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

Input

```
lst=[]
n=int(input("Enter an integer:"))
print("Integer numbers are")
for i in range(0,n):
    j=int(input())
    if j>100:
    lst.append("over")
   else:
    lst.append(j)
print(lst)
```

Output

```
stud@debian:~/anagha18$ python3 5.py
Enter an integer:5
Integer numbers are
 12
37
112
256
2
[12, 37, 'over', 'over', 2]
```

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

Input

```
list1=["ammu","appu","athira"]
count=0
for word in list1:
   for letter in word:
     if letter=="a":
        count=count+1
 print("The occurences of 'a' within the list is "+str(count))
```

Output

```
stud@debian:~/anagha18$ python3 6.py
The occurences of 'a' within the list is 4
```

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

Input

```
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:~/anagha18$ python3 7a.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
```

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

Input

```
str1=input("Enter a string:")

print("Original string:",str1)

char=str1[0]

str1=str1.replace(char,'$')

str1=char+str1[1:]

print("String:",str1)
```

Output

```
stud@debian:~/anagha18$ python3 8a.py
Enter a string:onion
Original string: onion
String: oni$n
```

8.Create a string from given string where first and last characters exchanged. [eg:

python ->nythop]

Input

```
s="python"
t=s[0]
t1=s[-1]
n=len(s)
rs=t1+s[1:n-1]+t
print(rs)
```

Output

```
stud@debian:~/anagha18$ python3 9a.py
nythop
```

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

Input

```
r=int(input("Enter the radius:"))

a=3.14*r*r

print("Area of circle is",a)
```

Output

```
stud@debian:~/anagha18$ python3 10.py
Enter the radius: 5
Area of circle is 78.5
```

10.Find biggest of 3 numbers entered.

Input

```
a=int(input("Enter the first number:"))

b=int(input("Enter the second number:"))

c=int(input("Enter the third number:"))

if a>b:

   if a>c:

      print(a)

  else:

     print(c)

else:

  if b>c:

       print(b)

else:

     print(c)
```

Output

```
stud@debian:~/anagha18$ python3 11.py
Enter the first number: 24
Enter the second number: 56
Enter the third number:78
78
```

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

Input

```
import os

a=input("Enter the file name:")

print("The extension of file",a,"is",os.path.splitext(a))
```

Output

```
stud@debian:~/anagha18$ python3 12a.py
Enter the file name: 8a.py
The extension of file  8a.py is (' 8a', '.py')
```

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

Input

```
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:~/anagha18$ python3 13.py
Enter colors separated by comma:
green,red,yellow,white
First and last colors in the list are green and white
```

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

Input

```
x=int(input("Enter an integer:"))
n1=str(x)
n2=n1+n1
n3=n2+n1
result=int(n1)+int(n2)+int(n3)
print(result)
```

Output

```
stud@debian:~/anagha18$ python3 14.py
Enter an integer:5
615
```

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

Input

```
l1=["red","green","blue","maroon","peach","orange"]

l2=["white","green","maroon","black"]

print(l1)

print(l2)

for i in l1:

  if i not in l2:

    print(i)
```

Output

```
stud@debian:~/anagha18$ python3 15.py
['red', 'green', 'blue', 'maroon', 'peach', 'orange']
['white', 'green', 'maroon', 'black']
red
blue
peach
orange
```

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

Input

```
str1=input("Enter first string:")

str2=input("Enter second string:")

str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]

print(str3)
```

Output

```
Enter first string:apple
Enter second string:orange
opple arange
```

16.Sort dictionary in ascending and descending order.

Input

```
dict1={"a":1,"c":3,"d":2,"b":4}

l=list(dict1.items())

print(l)

l.sort()

print("Ascending order is\n",l)

l=list(dict1.items())

l.sort(reverse=True)

print("Descending order is\n",l)
```

Output

```
stud@debian:~/anagha18$ python3 17.py
ascending order [('ammu', 1), ('baby', 2), ('christy', 3)]
descending order [('christy', 3), ('baby', 2), ('ammu', 1)]
```

17. Merge two dictionaries.

Input

```
dict1={"Name":"Athidhi","Age":25}

dict2={"Gender":"F","Qualification":"PG"}

dict1.update(dict2)

print(dict1)
```

Output

```
stud@debian:~/anagha18$ python3 merge1.py
{'Name': 'Anagha', 'Age': 15, 'Gender': 'F', 'Qualification': 'PG'}
```

18.Find gcd of 2 numbers.

Input

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
if x>y:
   s=y
else:
   s=x
for i in range(1,s+1):
   if(x%i==0)and(y%i==0):
      hcf=i
  print(hcf)
```

Output

```
stud@debian:~/anagha18$ python3 gcd1.py
Enter first number:24
Enter second number:8
8
```

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

Input

```
l1=[1,2,3,4,5,6]

l2=[]

for i in l1:

   if i%2!=0:

l2.append(i)

print(l2)
```

Output

```
stud@debian:~/anagha18$ python3 p20.py
[1, 3, 5]
```

COURSE OUTCOME 2

1. Program to find the factorial of a number.

Input

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

Output

```
stud@debian:~/anagha18$ python3 coa.py
Enter a number 5
120
```

2. Generate Fibonacci series of N terms.

Input

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

Output

```
stud@debian:~/anagha18$ python3 fibonacci.py
Enter a number 5
0
1
1
2
3
```

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

Input

```
list1=[1,2,3,4,5]
sum=0
for i in list1:
  sum=sum+i
  print(sum)
```

output

```
stud@debian:~/anagha18$ python3 co3.py
15
```

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

Input

```
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:~/anagha18$ python3 co4.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

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

Eg: N=4 1

2 4

3 6 9

4 8 12 16

Input

```
n=int(input("Enter a number:"))
 for i in range(1,n+1):
  for j in range(i,(i*i)+1,i):
    print(j,"\t",end="")
print("\n")
```

Output

```
stud@debian:~/anagha18$ python3 co5.py
Enter a number: 4
1

2       4

3       6       9

4       8       12      16
```

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

Input

```
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:~/anagha18$ python3 co6.py
Enter a string:anagha
a       : 3
n       : 1
g       : 1
h       : 1
```

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

Input

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

Output

```
stud@debian:~/anagha18$ python3 co7.py
Enter a string:increasing
increasingly
```

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

Input

```
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:~/anagha18$ python3 co8.py
Enter the range:5
Enter the words:
ammu
appu
bad
something
display
Length of longest word is 9
```

9. Construct following pattern using nested loop.

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

Input

```
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

```
stud@debian:~/anagha18$ python3 co9.py
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

10. Generate all factors of a number.

Input

```
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:~/anagha18$ python3 co10.py
Enter a number:24
Factors are
1
2
3
4
6
8
12
24
```

COURSE OUTCOME 3

PROGRAM 1:

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 modules and import * statements

Graphice\circle.py

```
from math import pi
def
    area_circle(radius):
    returnpi*radius*ra
    dius 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 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

```
PS D:\mySpace\learn> cd python
PS D:\mySpace\learn\python> md Graphics

    Directory: D:\mySpace\learn\python

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         28-02-2022  08.29 PM                Graphics


PS D:\mySpace\learn\python> cd Graphics
PS D:\mySpace\learn\python\Graphics> notepad __init__.py
PS D:\mySpace\learn\python\Graphics> notepad circle.py
PS D:\mySpace\learn\python\Graphics> notepad rectangle.py
PS D:\mySpace\learn\python\Graphics> md tdgraphics

    Directory: D:\mySpace\learn\python\Graphics

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         28-02-2022  08.32 PM                tdgraphics


PS D:\mySpace\learn\python\Graphics> cd tdgraphics
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad __init__.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad cuboid.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> notepad sphere.py
PS D:\mySpace\learn\python\Graphics\tdgraphics> cd ..
PS D:\mySpace\learn\python\Graphics> cd ..
PS D:\mySpace\learn\python>
```

```
PS D:\mySpace\learn\python> python graphics.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
PS D:\mySpace\learn\python>
```

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.

Program:

```
Class Rectangle:
def init (self,l,b):
        self.l=l
        self.b=b
def area(self):
        return (self.l*self.b)
def perimeter(self):
        return (2*(self.l+self.b))
def print(self):
        print(self.area)
r1=Rectangle(10,2)
r2=Rectangle(5,8)
x=r1.area()
y=r2.area()
print("area of first rectangle is",x)
print("area of second rectangle is",y)
p=r1.perimeter()
q=r2.perimeter()
print("perimeter of first rectangle is",p)
print("perimeter of second rectangle is",q)
print
if(x>y):
```

```
        print('Area of firstrectangle greater than second rectangle');
else:
        print('Area of second rectangle greater than first rectangle');
```

Output

```
area of first rectangle is 20
area of second rectangle is 40
perimeter of first rectangle is 24
perimeter of second rectangle is 26
Area of second rectangle greater than first rectangle
```

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

Program:

```
class Bank:
        def init (self,accno,aname,a_type,bal):
                self.accno=accno
                self.aname=aname
                self.a_type=a_type
                self.bal=bal
            def withdraw(self,x):
                    self.bal=self.bal-x
            def deposit(self,y):
                    self.bal=self.bal+y
            def print(self):
         print(self.accno,self.aname,self.a_type,self.bal)
acc1=Bank(2435,'anju','sbi',10000)
acc2=Bank(5436,'aju','federal',22000)
acc1.withdraw(1000)
acc1.deposit(4000)
acc2.withdraw(2500)
acc1.deposit(8000)
acc1.print()
acc2.print()
```

Output

```
2435 anju sbi 21000
5436 aju federal 19500
```

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

Program:

```
class Rectangle:
        def_init_(self,ln,br):
                self.ln=ln
                self.br=br
          def area(self):
                 p=self.ln*self.br
                 return p
          def_lt_(self,r1):
                if r2.area()<r1.area():
                return r2.area()
                else:
                return r1.area()

         def perim(self):
                q=2*(self.ln+self.br)
                return q
a=int(input("Enter length of the first rectangle:"))
b=int(input("Enter breadth of the first rectangle:"))
r1=Rectangle(a,b)
a=int(input("Enter length of the second rectangle:"))
b=int(input("Enter breadth of the second rectangle:"))
r2=Rectangle(a,b)
print("Perimeter of first rectangle= ",r1.perim())
print("Perimeter of second rectangle= ",r2.perim())
print("Least one is:",r1<r2)
```

Output:

```
Enter length of the first rectangle:12
Enter breadth of the first rectangle:4
Enter length of the second rectangle:8
Enter breadth of the second rectangle:6
Perimeter of first rectangle=  32
Perimeter of second rectangle=  28
Least one is: 48
```

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

Program:

```
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:

```
7 : 65 : 59
```

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 overreading.

Input

```
class Publisher:
    def__init__(self,name):
        self.name=name
class Book(Publisher):
    def__init__(self,name,title,auther):
        super().__init__(name)
        self.title=title
        self.auther=auther
    def print_function(self):
print("This Fuction is a member fuction of class Publisher")
class Python(Book):
    def__init__(self,name,title,auther,price,nop):
        super().__init__(name,title,auther)
            self.price=price
            self.nop=nop
    def print_function(self):
print("Name",self.name)
print("Title";,self.title)
print("Auther";,self.auther)
print("Price",self.price)
print("Number of Pages";,self.nop)
```

```
p1=Python(“Text book”,”PythonProgramming”;”Mr.abc&”,100,500)
p1.print_function()
p2=Book(“a”,”b”,”c”)
p2.print_function()
```

OUTPUT

```
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 100
Number of Pages : 500
This Fuction is a member fuction of class Publisher
```

COURSE OUTCOME 5

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

Program:

```
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.

Program:

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

Output:

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn2.py
['Name', 'Designation', 'Salary']
['Jessy', 'Manager', '90000']
['Tom', 'Clerk', '40000']
['Alfred', 'Assistant Manager', '70000']
PS C:\Users\HP\OneDrive\Desktop\python\co5>
```