

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-22



20 MCA 132 PROGRAMMING LABORATORY RECORD

Submitted by

NISHAN

REG NO: SNG21MCA-2026

in partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATIONS

**SREE NARAYANA GURUKULAM COLLEGE OF
ENGINEERING KADAYIRUPPU, KOLENCHERY 682 311**

(Affiliated to APJ Abdul Kalam Technological University)



20 MCA 132 PROGRAMMING LABORATORY RECORD

*Certified that this is a Bonafide record of practical work done by **Nishan** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree in Master of Computer Applications of Sree Narayana Gurukulam College of Engineering done during the Academic year 2021-22.*

Kadayiruppu

Course Instructor

Date:

Head of the Department

Prof.Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2026 **on** -----

External Examiner

Internal Examiner

SL NO.	DATE	NAME OF EXPERIMENT	PAGE NO.	REMARK
I	CO1			
1	24/11/21	Familiarizing Text Editor, IDE, Code Analysis Tools etc	1	
2	24/11/21	Leap Year	2	
3	24/11/21	List comprehensions	3	
4	24/11/21	occurrences of each word	5	
5	24/11/21	Prompt the user for a list of integers.	6	
6	24/11/21	Store a list of first names.	7	
7	24/11/21	Checking list are of same length,sums to same value,any value occur in both	8	
8	24/11/21	Get a string from an input string and replacing a character	10	
9	24/11/21	Create a string from given string where first and last characters exchanged.	11	
10	24/11/21	Accept the radius from user and find area of circle	12	
11	29/11/21	Find biggest of 3 numbers entered	13	
12	29/11/21	Accept a file name from user and print extension of that	14	
13	29/11/21	Create a list of colors,Display first and last colors.	15	
14	29/11/21	Accept an integer n and compute n+nn+nnn	16	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	17	
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	18	
17	29/11/21	Sort dictionary in ascending and descending order	19	
18	29/11/21	Merge two dictionaries	20	
19	29/11/21	Find gcd of 2 numbers.	21	

20	29/11/21	From a list of integers, create a list removing even numbers.	22	
II	CO2			
1	1/12/21	Program to find the factorial of a number	23	
2	1/12/21	Generate Fibonacci series of N terms	24	
3	1/12/21	Find the sum of all items in a list	25	
4	1/12/21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	26	
5	1/12/21	Display the given pyramid with step number accepted from user	27	
6	1/12/21	Count the number of characters (character frequency) in a string	28	
7	8/12/21	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	29	
8	8/12/21	Accept a list of words and return length of longest word	30	
9	8/12/21	Construct pattern using nested loop	31	
10	8/12/21	Generate all factors of a number. def print_factors(x):	32	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	33	
III	CO3			
1	15/12/21	Work with built-in packages	34	
2	15/12/21	Creation of packages	41	
IV	CO4			
1	9/1/22	Compare two Rectangle objects by their area	43	
2	9/1/22	Create a Bank account with members account number, name, type of account and balance.	45	
3	9/1/22	Overload '<' operator to compare the area of 2 rectangles.	47	

4	9/1/22	Overload '+' operator to find sum of 2 time	49	
5	9/1/22	Use base class constructor invocation and method overriding.	51	
V	CO5			
1	30/1/22	Write a Python program to read a file line by line and store it into a list.	52	
2	30/1/22	Python program to copy odd lines of one file to other	53	
3	30/1/22	Write a Python program to read each row from a given csv file and print a list of strings.	54	
4	30/1/22	Write a Python program to read specific columns of a given CSV file	55	
5	30/1/22	Write a Python program to write a Python dictionary to a csv file.	56	

I. COURSE OUTCOME 1(CO1)

PROGRAM NO: 1

DATE:24/11/2021

AIM: Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,... And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs.

Comparison between IDLE and Thonny

Thonny is built for education and you can download the latest version from the Thonny website. The download options are at the top right. Thonny looks quite different to IDLE - it has different panels for the editor, the shell and the variables watcher plus (show view) lots of other options as well. It has a powerful debugger built in and other tools which let you manage packages and plugins.

The IDLE editor comes built-in with Python and is the one that many tutorials use by default. It's a fine, basic, editor that also has a Python shell built in for interactive programming. When you start Idle up, you get the shell window. This allows you to execute python commands and see the results immediately without having to create a program. This can be useful for trying things out.

PROGRAM NO: 2

DATE:24/11/2021

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

```
s=int(input("enter start year:"))
e=int(input("enter end year:"))
if(s<e):
    print("leap years are:",end=" ")
for i in range(s,e):
    if i%4==0 and i%100!=0:
        print(i,end=" ")
```

OUTPUT

```
>>> %Run leapyear.py
enter start year:2020
enter end year:2030
leap years are: 2020 2024 2028
>>> |
```

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

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

```
list1 =[-10,20,35,-67,70]
re=[num for num in list1 if num>=0]
print(re)
```

OUTPUT

```
[20, 35, 70]
>>> |
```

- **Square of N number**

```
n=int(input("enter limit"))
l=[i**2 for i in range(0,n+1)]
print("square of N numbers ",l)
```

OUTPUT

```
enter limit:5
square of N numbers: [1, 4, 9, 16, 25]
>>> |
```


- **Form a list of vowels selected from a given word**

```
string=str(input("enter the word"))
print("the original string:",string)
print("the vowels are:",end=" ")
for i in string:
    if i in 'aeiouAEIOU':
        print([i],end=" ")
```

OUTPUT

```
enter the word:WONDERFUL
the original string is:WONDERFUL
the vowels are :['O']['E']['U']
>>> |
```

- **List ordinal value of each element of a word (Hint: use ord() to get ordinal values)**

```
w=input("enter a word:")
print("ordinalvalues corresponding to each element is:")
for i in w:
    print(i,end=":")
    print(ord(i),end=" ")
```

OUTPUT

```
enter a word:WONDERFUL
ordinal values corresponding to each element is
W:87 O:79 N:78 D:68 E:69 R:82 F:70 U:85 L:76
>>> |
```

PROGRAM NO: 4

DATE:24/11/2021

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

```
str1 = input("enter a string : ")
wordlist = str1.split()
count= []
for w in wordlist:
    count.append(wordlist.count(w))
print("count of the occurrence:" + str(list(zip(wordlist, count))))
```

OUTPUT

```
enter a stringIAM COMING HOME
count of occurrence:[('IAM', 1), ('COMING', 1), ('HOME', 1)]
>>> |
```

PROGRAM NO: 5

DATE:24/11/2021

AIM: Prompt the user for a list of integers. For all values greater than 100, store 'over' instead

```
n=[]
s=int(input("enter a limit:"))
print("enter {s} values")
for i in range(0,s):
    n.append(int(input()))
print("\nThe list after assinging:\n")
for i in range(0,len(n)):
    if n[i]>=100:
        print("over")
    else:
        print(n[i])
```

OUTPUT

```
enter a limit4
enter {s} values
23
56
99
101

the list is after asssigning:

23
56
99
over
>>> |
```

PROGRAM NO: 6

DATE:24/11/2021

AIM: Store a list of first names. Count the occurrences of 'a' within the list

```
a_list = ["a", "b", "a"]  
occ = a_list.count("a")  
print("count of occurrences of a :",occ)
```

OUTPUT

```
count of occurnces of a : 2  
>>> |
```

PROGRAM NO: 7

DATE:24/11/2021

AIM: 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

```
lst=[1,3,5,7,9,11,34]
lst1=[5,13,45,7,20,65,1]
s=int(0)
c=int(0)
if len(lst)==len(lst1):
    print("Lists are of same length")
else:
    print("Lists have different length")
for i in range(0,len(lst) and len(lst1)):
    s=s+lst[i]
    c=c+lst1[i]
if(s==c):
    print("equal sum")
else:
    print("not same sum")
print("elements that matched are:")
l=[]
for i in range(0,len(lst)):
    for j in range(0,len(lst1)):
        if lst[i]==lst1[j]:
            l.append(lst[i] and lst1[j])
        else:
            continue
print(l)
```

OUTPUT

```
lists are of same length  
not same sum  
elements that matched are:  
[1, 5, 7]  
>>> |
```

PROGRAM NO: 8

DATE:24/11/2021

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

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

OUTPUT

```
enter a string:MALAYALAM
MALAYALA$
>>> |
```

PROGRAM NO: 9

DATE:24/11/2021

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

```
str=input("enter a string")
new_str=str[-1:]+str[1:-1]+str[:1]
print("new string:",new_str)
```

OUTPUT

```
enter a stringPYTHON
new string: NYTHOP
>>> |
```


PROGRAM NO: 10

DATE:24/11/2021

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

```
pi=3.14
r=float(input("input the radius of circle:"))
result=3.14*r**2
print("the area of circLe with radius is:",result)
```

OUTPUT

```
input the radius of circle:2
the area of circLe with radius is: 12.56
>>> |
```

PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

```
x = int(input("enter 1st number: "))
y = int(input("enter 2nd number: "))
z = int(input("enter 3rd number: "))
if (x > y) and (x > z):
    largest = x
elif (y > x) and (y > z):
    largest = y
else:
    largest = z
print("largest number is",largest)
```

OUTPUT

```
enter 1st number56
enter 2nd number:45
enter 3rd number99
largest no is 99
>>> |
```

PROGRAM NO: 12

DATE:29/11/2021

AIM: Accept a file name from user and print extension of that

```
str= input("enter filename : ")
file=str.split(".")
print("extension of file is : " + file[-1])
```

OUTPUT

```
enter filename:AABBB.JAVA
extension of file is:JAVA
>>> |
```

PROGRAM NO: 13

DATE:29/11/2021

AIM: Create a list of colours from comma-separated colour names entered by user. Display first and last colours.

```
a=[]
for i in range(3):
    b=input("enter the color:")
    a.append(b)
print(a)
print(a[0])
print(a[2])
```

OUTPUT

```
enter the colorRED
enter the colorWHITE
enter the colorBLUE
['RED', 'WHITE', 'BLUE']
RED
BLUE
>>> |
```

PROGRAM NO: 14

DATE:29/11/2021

AIM: Accept an integer n and compute n+nn+nnn

```
n = int(input("enter a number : "))
x = int("%s" % n)
y = int("%s%s" %(n,n))
z = int("%s%s%s" %(n,n,n))
print ("n+nn+nnn:",x+y+z)
```

OUTPUT

```
enter a number:5
n+nn+nnn: 615
>>> |
```

PROGRAM NO: 15

DATE:29/11/2021

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

```
color_list_1 = set(["White", "pink", "Red","Blue"])  
color_list_2 = set(["Red", "Green","pink"])  
print(color_list_1.difference(color_list_2))
```

OUTPUT

```
>>> %Run listcollurss.py  
{'White', 'Blue'}
```

PROGRAM NO: 16**DATE:29/11/2021**

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

```
a="python"  
b="java"  
p1=a[0]  
p2=b[0]  
c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]  
print(c)
```

OUTPUT

```
jython pava  
>>>
```

PROGRAM NO: 17

DATE:29/11/2021

AIM: Sort dictionary in ascending and descending order.

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('original dictionary : ',d)
sorted_d=sorted(d.items(), key=operator.itemgetter(1))
print('dictionary in ascending order by value ',sorted_d)
sorted_d =dict(sorted(d.items(), key=operator.itemgetter(1),reverse=True))
print('dictionary in descending order by value : ',sorted_d)
```

OUTPUT

```
original dictionary: {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
dictionary in acending order by value [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]
dictionary in descending order by value: {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
>>> |
```


PROGRAM NO: 18

DATE:29/11/2021

AIM: Merge two dictionaries

```
d1 ={'a': 100, 'b': 200}
d2 ={'x' : 300, 'y': 200}
print ("dict ionary 1=:", d1)
print ("dictionary 2-: ", d2)
d =d1. copy ()
d.update (d2)
print ("merged dictionary: ", d)
```

OUTPUT

```
dictionary1=: {'a': 100, 'b': 200}
dictionary2=: {'x': 300, 'y': 200}
merged dictionary: {'a': 100, 'b': 200, 'x': 300, 'y': 200}
>>> |
```

PROGRAM NO: 19

DATE:29/11/2021

AIM: Find gcd of 2 numbers.

```
x= int(input("enter 1st no: "))
y= int(input("enter 2nd no: "))
i = 1
while(i <= x and i <= y):
    if(x % i == 0 and y% i == 0):
        gcd = i
        i = i + 1
print("gcd: ", gcd)
```

OUTPUT

```
enter 1st no120
enter 2nd no5
gcd: 5
>>> |
```

PROGRAM NO: 20

DATE:29/11/2021

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

```
num = [7,8, 120, 25, 44, 20, 27]
print( "original list:",num)
num = [x for x in num if x%2!=0]
print("list after removing even nos:",num)
```

OUTPUT

```
original list: [7, 8, 120, 25, 44, 20, 27]
list after remooving even nos [7, 25, 27]
>>> |
```

II COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE:1/12/2021

AIM: Program to find the factorial of a number

```
n=int(input('enter a number : '))
f=1
for i in range(1,n+1):
    f=f*i
print ('factorial of ',n, '=',f)
```

OUTPUT

```
>>>
enter a number5
factorial of 5 = 120
>>> |
```

PROGRAM NO: 2

DATE:1/12/2021

AIM: Generate Fibonacci series of N terms

```
n = int(input("enter the limit : "))
a = 0
b = 1
sum = 0
count = 1
print("fibonacci series :",end= " ")
while(count <= n):
    print(sum, end = " ")
    count += 1
    a = b
    b = sum
    sum = a + b
```

OUTPUT

```
>>>
enter the limit6
fibonacci series0 1 1 2 3 5
>>> |
```

PROGRAM NO: 3

DATE:1/12/2021

AIM: Find the sum of all items in a list

```
list1 = [10, 20, 30, 40, 50]
```

```
total = sum(list1)
```

```
print("sum of list : ",total)
```

OUTPUT

```
>>>  
sum of list: 100  
>>> |
```

PROGRAM NO: 4

DATE:1/12/2021

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

```
from math import sqrt as s
for i in range(1000,10000):
    if s(i)==int(s(i)) and i%2==0:
        print(i,end=" ")
```

OUTPUT

```
>>>
1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 3136 3364 3600 3844
4096 4356 4624 4900 5184 5476 5776 6084 6400 6724 7056 7396 7744 8100 8464 8836
9216 9604
>>> |
```

PROGRAM NO: 5

DATE:1/12/2021

AIM: Display the given pyramid with step number accepted from user

```
rows = int(input("enter a number of rows"))
for i in range(1, rows+1):
    for j in range(1,i+1):
        print(i * j, end=' ')
    print()
```

OUTPUT

```
>>>
enter a number of rows3
1
24
369
>>> |
```


PROGRAM NO: 6

DATE:1/12/2021

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

```
test_str=str(input("enter the string : "))
freq = {}
for i in test_str:
    if i in freq:
        freq[i] += 1
    else:
        freq[i] = 1
print ("count of all characters : "+ str(freq))
```

OUTPUT

```
>>>
enter the stringmalayalam
count of all characters: {'l': 2, 'a': 4, 'y': 1, 'm': 2}
>>> |
```

PROGRAM NO: 7

DATE:8/12/2021

AIM: Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'

```
str=input("enter a string")
print("inputted string is ",str)
if(str.endswith("ing")):
    str=str+'ly'
else:
    str=str+'ing'
print("the formatted string is  ",str)
```

OUTPUT

```
>>>
enter a stringswim
inputted string is  swim
the formatted string is swimming
>>> |
```

PROGRAM NO: 8

DATE:8/12/2021

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

```
a=[]
n= int(input("enter the number of elements in list : "))
for x in range(0,n):
    element=input("enter element "+ str(x+1) )
    a.append(element)
    max1=len(a[0])
    temp=a[0]
for i in a:
    if(len(i)>max1):
        max1=len(i)
        temp=i
print("longest Word : ",temp)
print("length of longest word : ",max1)
```

OUTPUT

```
>>>
enter the  number of elements in list2
enter element1python
enter element2programming
longest word programming
length of longest word: 11
>>> |
```

PROGRAM NO: 9

DATE:8/12/2021

AIM: Construct following pattern using nested loop

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

```
n=int(input("enter the limit:"))
for i in range(n+1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
for i in range(n,0,-1):
    for j in range(i):
        print("*",end=" ")
    print("\n")
```

OUTPUT

```
>>>
enter the limit4

*
**
***
****
*****
****
***
**
*
>>> |
```

PROGRAM NO:10

DATE:8/12/2021

AIM: Generate all factors of a number. def print_factors(x):

```
def factors(x):  
    print("the factors of",x,"are")  
    for i in range(1, x + 1):  
        if x % i == 0:  
            print(i)  
n=int(input("enter a number:"))  
factors(n)
```

OUTPUT

```
>>>  
enter a number:16  
the factors of 16 are  
1  
2  
4  
8  
16  
>>> |
```

PROGRAM NO:11

DATE:8/12/2021

AIM: Write lambda functions to find area of square, rectangle and triangle.

```
import math
tarea = lambda b,h : 1/2*b*h
rarea = lambda l,b : l*b
sarea = lambda a : a*a

print("area of triangle ", tarea(20,60))
print("area of rectangle", rarea(10,30))
print("area of square ", sarea(14))
```

OUTPUT

```
area of triangle 100.0
area of rectangle 600
area of square 4
>>> |
```

III. COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

DATE:15/12/2021

AIM:Design modules and packages – builtin and user defined packages.

A)MATH MODULE

```
import math
print("the value of pi",math.pi)
import math as m
print("the value of pi is ", m.pi)
from math import pi,sqrt
print("the value of pi is  ", pi)
print("the square root of 4 is  ", sqrt(4))

print(math.cos(90))
print(math.sin(3))
print(math.tan(0))
```

OUTPUT

```
the value of pi is 3.141592653589793
the value of pi is 3.141592653589793
the value of pi is  3.141592653589793
the square root of 4 is  2.0
-0.4480736161291701
0.1411200080598672
0.0
>>> |
```

B)TIME MODULE

```
import time
print("current time in sec:",time.time())
print("current time:",time.ctime())
print("current time after 30 sec:",time.ctime(time.time()+30))
t=time.localtime()
print("time:",t)
print("current Year:",t.tm_year)
print("current Month:",t.tm_mon)
print("current Day:",t.tm_mday)
print("current Hour:",t.tm_hour)
print("current week:",t.tm_wday)
print("Day of Year:",t.tm_yday)
print("formatted time:",time.strftime("%d /%m /%y    %H:%M:%S    %Y ",t))
print("formatted time:",time.strftime("%c    %H:%M:%S    %Y ",t))
print(".....")
```

OUTPUT

```
current time in sec: 1639963084.789859
current time: Mon Dec 20 06:48:04 2021
current time after 30 sec: Mon Dec 20 06:48:34 2021
time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=6, tm_min=48
, tm_sec=4, tm_wday=0, tm_yday=354, tm_isdst=0)
current Year: 2021
current Month: 12
current Day: 20
current Hour: 6
current week: 0
Day of Year: 354
formatted time: 20 /12 /21    06:48:04    2021
formatted time: Mon Dec 20 06:48:04 2021    06:48:04    2021
.....
>>> |
```


C)CALENDAR MODULE

```
import calendar
mm = int(input("Enter Month : "))
yy = int(input("Enter Year : "))
print(calendar.month(yy,mm))
print(calendar.calendar(2015))
```

OUTPUT

```
Enter month: 12
Enter year: 2021
December 2021
Mo Tu We Th Fr Sa Su
    1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

2015

January February March
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
    1  2  3  4          1          1
 5  6  7  8  9 10 11  2  3  4  5  6  7  8  9 10 11 12 13 14 15
12 13 14 15 16 17 18 16 17 18 19 20 21 22 16 17 18 19 20 21 22
19 20 21 22 23 24 25 23 24 25 26 27 28 23 24 25 26 27 28 29
26 27 28 29 30 31    30 31

April May June
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
    1  2  3  4  5          1  2  3          1  2  3  4  5  6  7
 6  7  8  9 10 11 12  4  5  6  7  8  9 10  8  9 10 11 12 13 14
13 14 15 16 17 18 19 11 12 13 14 15 16 17 15 16 17 18 19 20 21
20 21 22 23 24 25 26 18 19 20 21 22 23 24 22 23 24 25 26 27 28
27 28 29 30 25 26 27 28 29 30 31 29 30
```

D)RANDOM MODULE

```
import random
list1 = [1, 2, 3, 4, 5, 6]
print(random.choice(list1))
print("*****")
import random
random.seed(5)
print(random.random())
print(random.random())
print("*****")
import random
r1 = random.randint(5, 15)
print("Random number between 5 and 15 is % s" % (r1))
r2 = random.randint(-10, -2)
print("Random number between -10 and -2 is % d" % (r2))
print("*****")
import random
list1 = [1, 2, 3, 4, 5, 6]
print(random.choice(list1))
string = "geeks"
print(random.choice(string))
tuple1 = (1, 2, 3, 4, 5)
print(random.choice(tuple1))
```

OUTPUT

```
4
*****
0.6229016948897019
0.7417869892607294
*****
Random number between 5 and 15 is 15
Random number between -10 and -2 is -2
*****
1
k
2
>>>
```

E)STATISTICS MODULE

```
import statistics
list1 = [5,2,5,6,1,2,6,7,2,6,3,5,5]
x = statistics.mean(list1)
print(x)
y = statistics.median(list1)
print(y)
z = statistics.mode(list1)
print(z)
a = statistics.stdev(list1)
print(a)

b = statistics.variance(list1)
print(b)
```

OUTPUT

```
4.230769230769231
5
5
1.9644272343292228|
3.858974358974359
>>>
```

F)DATETIME MODULE

```
import datetime
t=datetime.time(22,56,44)
print(t)
print("Hour:",t.hour)
print("Minute:",t.minute)
print("Second:",t.second)
print("microsecond:",t.microsecond)
print(".....")

d=datetime.date.today()
print(d)
print("Year:",d.year)
print("Month:",d.month)
print("Day:",d.day)
print("timeandcalmodule.py")

d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2)
print(td)
d2=d1+td
print(d2)
print("d2-d1=",d2-d1)
print("d2>d1=",d2>d1)
print(".....")

d1=datetime.date.today()
t1=datetime.time(12,55,0)
dt=datetime.datetime.combine(d1,t1)
print('dt:',dt)
```

OUTPUT:

```
22:56:44
Hour: 22
Minute: 56
Second: 44
microsecond: 0
.....
2021-12-20
Year: 2021
Month: 12
Day: 20
timeandcalmodule.py
2021-12-20
2 days, 0:00:00
2021-12-22
d2-d1= 2 days, 0:00:00
d2>d1= True
.....
dt: 2021-12-20 12:55:00
>>> |
```

PROGRAM NO: 2

DATE:15/12/2021

AIM: 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)

graphicsuse.py

```
from graphics import rectangle
from graphics import circle
from graphics.ThreeDgraphics import cuboid
from graphics.ThreeDgraphics import sphere
l=int(input("Enter the length,l : "))
b=int(input("Enter the breadth,b : "))
rectangle.perimeter(l,b)
rectangle.area(l,b)
r=int(input("Enter the radius,r : "))
circle.perimeter(r)
circle.area(r)
l=int(input("Enter the length,l : "))
b=int(input("Enter the breadth,b : "))
h=int(input("Enter the height,h : "))
cuboid.perimeter(l,b,h)
cuboid.area(l,b,h)
r=int(input("Enter the radius,r : "))
sphere.volume(r)
sphere.area(r)
```

Package graphics

circle.py

```
def perimeter(r):
    print ("Perimeter : ",2*3.14*r)
def area(r):
    print ("Area : ",3.14*r*r)
```

rectangle.py

```
def perimeter(l,b):  
    print ("Perimeter : ",2*(l+b))  
def area(l,b):  
    print ("Area : ",l*b)
```

Subpackages:ThreeDgraphics

cuboid.py

```
def perimeter(l,b,h):  
    print ("Perimeter : ",4*(l+b+h))  
def area(l,b,h):  
    print ("Area : ",2*l*b+2*l*h+2*h*b)
```

sphere.py

```
def volume(r):  
    print ("Volume : ",(4/3)*3.14*r*r*r)  
def area(r):  
    print ("Surface Area : ",4*3.14*r*r)
```

OUTPUT

```
>>> %Run graphicsuse.py  
Enter the length,l : 4  
Enter the breadth,b : 5  
Perimeter : 18  
Area : 20  
Enter the radius,r : 5  
Perimeter : 31.400000000000002  
Area : 78.5  
Enter the length,l : 4  
Enter the breadth,b : 5  
Enter the height,h : 6  
Perimeter : 60  
Area : 148  
Enter the radius,r : 5  
Volume : 523.3333333333334  
Surface Area : 314.0  
>>> %Run graphicsuse.py
```

IV. COURSE OUTCOME 4(CO4)

PROGRAM NO: 1

DATE:9/1/2022

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

```
class rectangle():
    def __init__(self,breadth,length):
        self.breadth=breadth
        self.length=length
    def area(self):
        return self.breadth*self.length
    def perimeter(self):
        return 2*self.breadth+self.length

a1=int(input("Enter length of rectangle: "))
b1=int(input("Enter breadth of rectangle: "))

obj1=rectangle(a1,b1)

print("Area of rectangle:",obj1.area())
print("perimeter of rectangle:",obj1.perimeter())

a2=int(input("Enter length of rectangle: "))
b2=int(input("Enter breadth of rectangle: "))

obj2=rectangle(a2,b2)

print("Area of rectangle:",obj2.area())
print("perimeter of rectangle:",obj2.perimeter())

if (obj1.area())>obj2.area():
```



```
    print("obj1 is greater")
else:
    print("obj2 is greater")
```

OUTPUT

```
Enter length of rectangle: 5
Enter breadth of rectangle: 6
Area of rectangle: 30
perimeter of rectangle: 16
Enter length of rectangle: 9
Enter breadth of rectangle: 6
Area of rectangle: 54
perimeter of rectangle: 24
obj2 is greater
```

PROGRAM NO: 2

DATE:9/1/2022

AIM: 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.

```
class bank:
    def __init__(self):
        self.balance=0
        name=input("enter the name of account holder:")
        acno=int(input("enter the account no:"))

        print ("The account is created")
        print("\n name of account:",name)
        print("\n account no:",acno)

    def deposit(self):
        amount=int(input(" enter the amount:"))
        self.balance+=amount
    def withdraw(self):
        amount = float(input("Enter amount to be Withdrawn:"))

        if (self.balance>=amount):
            self.balance-=amount
            print("\nYou Withdraw:", amount)
        else:
            print("\ninsufficient balance")
    def display(self):
        print("\nAvailable Balance =",self.balance)

b=bank()
b.deposit()
b.withdraw()
b.display()
```

OUTPUT

```
Python 3.8.10 (/usr/bin/python3)
>>> %Run sebinton.py

enter the name of account holder:nishan
enter the account no:12345
The account is created

name of account: nishan

account no: 12345
enter the amount:5000
Enter amount to be Withdrawn:500

You Withdraw: 500.0

Available Balance = 4500.0
>>> |
```

PROGRAM NO: 3

DATE:9/1/2022

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

```
def __init__(self,length,width):  
    self.__length=length  
    self.__width=width  
def __lt__(self,a1):  
    area1=self.__length*self.__width  
    area2=a1.__length*a1.__width  
    if(area1<area2):  
        return(True)  
    else:  
        return(False)
```

```
a1=int(input("Length of 1 rectangle:"))  
b1=int(input("width 1 rectangle:"))  
r1=rectangle(a1,b1)
```

```
a2=int(input("Length 2 rectangle:"))  
b2=int(input("width 2 rectangle:"))  
r2=rectangle(a2,b2)  
if(r1<r2):  
    print("Rectangle 2 is larger!!")  
else:  
    print("Rectangle 1 is larger!!")
```

OUTPUT

```
Length of 1 rectangle:5  
width 1 rectangle:6  
Length 2 rectangle:9  
width 2 rectangle:6  
Rectangle 2 is larger!!
```

PROGRAM NO: 4

DATE:9/1/2022

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

class Time:

```
def __init__(self,hour,minute,second):
```

```
    self.__hour=hour
```

```
    self.__minute=minute
```

```
    self.__second=second
```

```
def __add__(self,h):
```

```
    second=self.__second+h.__second
```

```
    minute=self.__minute+h.__minute
```

```
    hour=self.__hour+h.__hour
```

```
    if(second>60):
```

```
        second=second-60
```

```
        minute=minute+1
```

```
    if(minute>60):
```

```
        minute=minute-60
```

```
        hour=hour+1
```

```
    return hour,minute,second
```

```
print("Enter 1 time:")
```

```
h1=int(input("enter the hour:"))
```

```
m1=int(input("enter the minute:"))
```

```
s1=int(input(" enter the second:"))
```

```
t1=Time(h1,m1,s1)
```

```
print("Enter 2 time:")
```

```
h2=int(input("enter the hour:"))
```

```
m2=int(input("enter the minute:"))
```

```
s2=int(input("enter the second:"))
```

```
t2=Time(h2,m2,s2)
```

```
hr,min,sec=t1+t2
```

```
print(hr,end=":")
```

```
print(min,end=":")
```

```
print(sec,end=" ")
```

OUTPUT

```
-----  
Enter 1 time:  
enter the hour:5  
enter the minute:66  
  enter the second:25  
Enter 2 time:  
enter the hour:6  
enter the minute:56  
enter the second:25  
12:62:50  
.
```

PROGRAM NO: 5

DATE:9/1/2022

AIM: 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.

```
class publisher:
    def getbook(self):
        self.title=input("title :")
        self.author=input("author :")

class python(publisher):
    def getdetails(self):
        self.price=int(input("price :"))
        self.nopages=int(input("nopages :"))

    def display(self):
        print("title of the book is :",self.title)
        print("author of the book is :",self.author)
        print("price is: ",self.price)
        print("number of pages :",self.nopages)

a=python()
a.getbook()
a.getdetails()
a.display()
```

OUTPUT

```
title :Qwerty
author :Manoj
price :100
nopages :1200
title of the book is : Qwerty
author of the book is : Manoj
price is:  100
number of pages : 1200
```


V. COURSE OUTCOME 5(CO5)

PROGRAM NO: 1

DATE:30/1/2022

AIM: Write a Python program to read a file line by line and store it into a list

```
f1=open("secfile.txt","w")
f1.write("This is my first file in python.\n want to work with files \n This is my third line")
f1=open("secfile.txt","r")
ff=f1.readlines()
print(ff)
```

OUTPUT

```
>>> %Run 1.py
['This is my first file in python.\n', ' want to work with files \n', ' This is my third line']
>>>
```

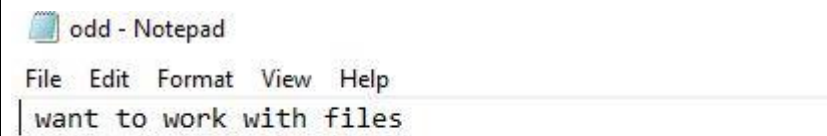
PROGRAM NO: 2

DATE:30/1/2022

AIM: Python program to copy odd lines of one file to other

```
f1=open("secfile.txt","r")
ff=f1.readlines()
with open("odd.txt","w") as f2:
    for x in range(0,len(ff)):
        if(x%2!=0):
            f2.write(ff[x])
```

OUTPUT



PROGRAM NO: 3

DATE:30/1/2022

AIM: Write a Python program to read each row from a given csv file and print a list of strings.

```
import csv
filename = "username.csv"
rows = []
cf=open(filename, 'r')
csvreader = csv.reader(cf)
for r in csvreader:
    rows.append(r)
print(rows)
cf.close()
```

OUTPUT

```
>>> %Run 3.py
[['Username', ' Identifier', 'Firstname', 'Lastname'], ['nicel', '3022', 'Neo', 'Anderson'], ['bue34', '2070', 'Ash', 'Grey'], ['priariel', '3456', 'Ben', 'steelwall'], ['jenkins46', '9346', 'Mary', 'Jenkins'], ['smith79', '5079', 'Jamie', 'Smith']]
>>>
```

PROGRAM NO: 4

DATE:30/1/2022

AIM: Write a Python program to read specific columns of a given CSV file and print the content of the columns.

```
import csv
filename = "emp.txt"
fields = []
rows = []
cf=open(filename, 'r')
csvreader = csv.DictReader(cf)
for r in csvreader:
    print(dict(r))
```

OUTPUT

```
>>> %Run C05_4.py
{'name': 'John Smith', 'department': 'Accounting', 'birthday month': 'November'}
{'name': 'Erica Meyers', 'department': 'IT', 'birthday month': 'March'}
>>>
```

PROGRAM NO: 5

DATE:30/1/2022

AIM: Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

```
import csv

field_names = ['No', 'Company', 'Car Model']

cars = [
    {'No': 1, 'Company': 'Ferrari', 'Car Model': '488 GTB'},
    {'No': 2, 'Company': 'Porsche', 'Car Model': '918 Spyder'},
    {'No': 3, 'Company': 'Bugatti', 'Car Model': 'La Voiture Noire'},
    {'No': 4, 'Company': 'Rolls Royce', 'Car Model': 'Phantom'},
    {'No': 5, 'Company': 'BMW', 'Car Model': 'BMW X7'},
]

with open('Names1.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_names)
    writer.writeheader()
    writer.writerows(cars)

filename = "names1.csv"

cf=open("names1.csv", 'r')
rows=[]
csvreader = csv.reader(cf)
for r in csvreader:
    rows.append(r)
for r in rows:
    print(*r)
```

OUTPUT

```
>>> %Run 5.py
No Company Car Model
1 Ferrari 488 GTB
2 Porsche 918 Spyder
3 Bugatti La Voiture Noire
4 Rolls Royce Phantom
5 BMW BMW X7
>>>
```