

SREENARAYANAGURUKULAM 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

AKHIL PAUL SAJI

REG NO: SNG21MCA-2003

in partial fulfillment for the award of the degree

in

MASTER OF COMPUTER APPLICATIONS

SREENARAYANAGURUKULAM 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 **AKHIL PAUL SAJI** 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-2003 On:

External Examiner

Internal Examiner

INDEX

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	3	
3	24/11/21	List comprehensions	4	
4	24/11/21	occurrences of each word	6	
5	24/11/21	Prompt the user for a list of integers.	7	
6	24/11/21	Store a list of first names.	8	
7	24/11/21	Checking list are of same length,sums to same value,any value occur in both	10	
8	24/11/21	Get a string from an input string and replacing a character	11	
9	24/11/21	Create a string from given string where first and last characters exchanged.	12	
10	24/11/21	Accept the radius from user and find area of circle	13	
11	29/11/21	Find biggest of 3 numbers entered	14	
12	29/11/21	Accept a file name from user and print extension of that	15	
13	29/11/21	Create a list of colors,Display first and last colors.	16	
14	29/11/21	Acceptan integern and compute n+nn+nnn	17	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	18	
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	19	

17	29/11/21	Sort dictionary in ascending & descending order	20	
18	29/11/21	Merge two dictionaries	21	
19	29/11/21	Find gcd of 2 numbers.	22	
20	29/11/21	From a list of integers, create a list removing even numbers.	23	
II	CO2			
1	1/12/21	Program to find the factorial of a number	24	
2	1/12/21	Generate Fibonacci series of N terms	25	
3	1/12/21	Find the sum of all items in a list	26	
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.	27	
5	1/12/21	Display the given pyramid with step number accepted from user	28	
6	1/12/21	Count the number of characters (character frequency) in a string	29	
7	8/12/21	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	30	
8	8/12/21	Accept a list of words and return length of longest word	31	
9	8/12/21	Construct pattern using nested loop	33	
10	8/12/21	Generate all factors of a number. def print_factors(x):	34	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	35	
III	CO3			
1	15/12/21	Work with built-in packages	41	

2	15/12/21	Creation of packages	45	
IV	CO4			
1	9/1/22	Compare two Rectangle objects by their area	47	
2	9/1/22	Create a Bank account with members account number, name, type of account and balance.	50	
3	9/1/22	Overload '<' operator to compare the area of 2 rectangles.	52	
4	9/1/22	Overload '+' operator to find sum of 2 time	54	
5	9/1/22	Use base class constructor invocation and method overriding.	56	
V	CO5			
1	30/1/22	Write a Python program to read a file line by line and store it into a list.	57	
2	30/1/22	Python program to copy odd lines of one file to other	58	
3	30/1/22	Write a Python program to read each row from a given csv file and print a list of strings.	59	
4	30/1/22	Write a Python program to read specific columns of a given CSV file	60	
5	30/1/22	Write a Python program to write a Python dictionary to a csv file.	62	

COURSE OUTCOME 1(CO1)

PROGRAM NO: 1

DATE:24/11/2021

AIM: Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev...

A text editor is a tool that allows a user to create and revise documents in a computer.

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

An IDE normally consists of at least a source code editor, build automation tools .

Top Python IDE's

- PyCharm
- Spyder
- Eclipse PyDev
- Wing
- IDLE

PyCharm

In industries most of the professional developers use PyCharm and it has been considered the best IDE for python developers. It was developed by the Czech company JetBrains and it's a cross-platform IDE.

- It is considered as an intelligent code editor, fast and safe refactoring, and smart code.

- Features for debugging, profiling, remote development, testing the code, auto code completion, quick fixing, error detection and tools of the database.
- Support for Popular web technologies, web frameworks, scientific libraries and version control.

COURSE OUTCOME 1(CO1)**PROGRAM NO: 2****DATE:24/11/2021**

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

Program Source Code:

```
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:

enter start year:2021

enter end year:2050

leap years are: 2024 2028 2032 2036 2040 2044 2048

COURSE OUTCOME 1(CO1)

PROGRAM NO: 3

DATE:24/11/2021

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

Program Source Code:

```
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:"))squarelist= [ i**2 for i in
range(1,n+1)] print("Square
of N numbers : ", squarelist)
```

OUTPUT:

enter limit:5

Square of N numbers : [1, 4, 9, 16, 25]

Form a list of vowels selected from a given word

```
word =str(input("Enter the word :"))
print("The original string is : "+word)
print("The vowel are : ",end="")
for i in word:
if i in 'aeiouAEIOU':
print([i],end=" ")
```

OUTPUT:

Enter the word :PYTHON PROGRAMMING

The original string is : PYTHON PROGRAMMING

The vowel are : ['O'] ['O'] ['A'] ['I']

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

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

OUTPUT:

Enter a word:PYTHON

Ordinal values corresponding to each element is:

P:80 Y:89 T:84 H:72 O:79 N:78

COURSE OUTCOME 1(CO1)**PROGRAM NO: 4****DATE:24/11/2021****AIM :** Count the occurrences of each word in a line of text.**Program Source Code:**

```
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 string : Python is a programming language  
count of the occurrence:[('Python', 1), ('is', 1), ('a', 1), ('programming', 1),  
('language', 1)]
```

COURSE OUTCOME 1(CO1)

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

Program Source Code:

```
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 limit:2
Enter {s} values
24
199
The list after assinging:
24
over
```

COURSE OUTCOME 1(CO1)

PROGRAM NO: 6

DATE:24/11/2021

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

Program Source Code:

```
a_list = ["a", "b", "a"]
```

```
occ = a_list.count("a")
```

```
print("count of occurrences of a :",occ)
```

OUTPUT:

count of occurrences of a : 2

COURSE OUTCOME 1(CO1)

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.

Program Source Code:

```
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(lst1)):  
    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]

COURSE OUTCOME 1(CO1)

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]

Program Source Code:

```
str1="malayalam"  
  
char = str1[0]  
  
str1 = str1.replace(char, '$')  
  
str1 = char + str1[1:]  
  
print(str1)
```

OUTPUT :

malayala\$

COURSE OUTCOME 1(CO1)**PROGRAM NO: 9****DATE:24/11/2021**

AIM : Create a string from given string where first and last characters exchanged.

Program Source Code:

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

OUTPUT:

Enter a string : python

New string : nythop

COURSE OUTCOME 1(CO1)**PROGRAM NO: 10****DATE:24/11/2021**

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

Program Source Code:

```
pi=3.14
```

```
r = float(input ("Input the radius of the circle : "))
```

```
result=3.14 * r**2
```

```
print ("The area of the circle with radius is: ", result)
```

OUTPUT:

Input the radius of the circle : 2

The area of the circle with radius is: 12.56

COURSE OUTCOME 1(CO1)

PROGRAM NO: 11

DATE:29/11/2021

AIM : Find biggest of 3 numbers entered.

Program Source Code:

```
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("The largest number is",largest)
```

OUTPUT:

Enter 1st number: 3

Enter 2nd number: 2

Enter 3rd number: 10

The largest number is 10

COURSE OUTCOME 1(CO1)

PROGRAM NO: 12

DATE:29/11/2021

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

Program Source Code:

```
file= input("Enter filename : ")  
f=file.split(".")  
print("Extension of the file is : " + f[-1])
```

OUTPUT:

Enter filename: abc.java

Extension of the file is : java

COURSE OUTCOME 1(CO1)

PROGRAM NO: 13

DATE:29/11/2021

AIM : Create a list of colors from comma-separated color names entered by user.

Display first and last colors.

Program Source Code:

```
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 colorgreen  
enter the colorblue  
enter the colorwhite  
['green', 'blue', 'white']  
green  
white
```

COURSE OUTCOME 1(CO1)

PROGRAM NO: 14

DATE:29/11/2021

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

Program Source Code:

```
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

COURSE OUTCOME 1(CO1)

PROGRAM NO: 15

DATE:29/11/2021

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

Program Source Code:

```
color_list_1 = set(["White", "pink", "Red", "Blue"])
```

```
color_list_2 = set(["Red", "Green", "pink"])
```

```
print(color_list_1.difference(color_list_2))
```

OUTPUT:

```
{'White', 'Blue'}
```

COURSE OUTCOME 1(CO1)

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.

Program Source Code:

```
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

COURSE OUTCOME 1(CO1)

PROGRAM NO: 17

DATE:29/11/2021

AIM : Sort dictionary in ascending and descending order.

Program Source Code:

```
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 ascending 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}

COURSE OUTCOME 1(CO1)

PROGRAM NO: 18

DATE:29/11/2021

AIM : Merge two dictionaries.

Program Source Code:

```
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:

Dictionary 1=: { 'a': 100, 'b': 200 }

Dictionary 2-: { 'x': 300, 'y': 200 }

Merged Dictionary: { 'a': 100, 'b': 200, 'x': 300, 'y': 200 }

COURSE OUTCOME 1(CO1)

PROGRAM NO: 19

DATE:29/11/2021

AIM : Find gcd of 2 numbers.

Program Source Code:

```
x= int(input("Enter 1st number: "))  
  
y= int(input("Enter 2nd number: "))  
  
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 number: 120

Enter 2nd number: 5

GCD : 5

COURSE OUTCOME 1(CO1)**PROGRAM NO: 20****DATE:29/11/2021****AIM :** From a list of integers, create a list removing even numbers.**Program Source Code:**

```
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 numbers:",num)
```

OUTPUT:

Original list: [7, 8, 120, 25, 44, 20, 27]

list after removing Even numbers: [7, 25, 27]

COURSE OUTCOME 2(CO2)

PROGRAM NO : 1

DATE: 1/12/2021

AIM : Program to find the factorial of a number.

Program Source Code:

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

OUTPUT :

```
enter the number3  
factorial of 3 is 6
```

COURSE OUTCOME 2(CO2)

PROGRAM NO : 2

DATE: 1/12/2021

AIM : Generate Fibonacci series of N.

Program Source Code:

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

OUTPUT :

```
enter the limit:5  
fibinocci series: 0 1 1 2 3 5
```

COURSE OUTCOME 2(CO2)**PROGRAM NO : 3****DATE: 1/12/2021****AIM :** Find the sum of all items in a list.**Program Source Code:**

```
list=[1,2,3,4,5,6,7,8,9,10]
```

```
total=sum(list)
```

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

OUTPUT :

```
sum of list= 55
```

COURSE OUTCOME 2(CO2)**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.

Program Source Code:

```
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

COURSE OUTCOME 2(CO2)

PROGRAM NO : 5

DATE: 1/12/2021

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

Program Source Code:

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

OUTPUT :

enter the number of rows4

1

2 4

3 6 9

4 8 12 16

COURSE OUTCOME 2(CO2)

PROGRAM NO : 6

DATE: 1/12/2021

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

Program Source Code:

```
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 string : hi

Count of all characters : {'h': 1, 'i': 1}

COURSE OUTCOME 2(CO2)**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'.

Program Source Code:

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

OUTPUT :

```
enter a string playing  
inputed string is playing  
the formated string is playingly
```

COURSE OUTCOME 2(CO2)

PROGRAM NO : 8

DATE: 8/12/2021

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

Program Source Code:

```
a=[]  
  
n=int(input("enter the number of elements in the 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 the list : 2  
enter element1 : malayalam  
enter element2 : maths  
longest word : malayalam  
length of longest word : 9
```

COURSE OUTCOME 2(CO2)

PROGRAM NO : 9

DATE: 8/12/2021

AIM : Construct following pattern using nested loop.

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

Program Source Code:

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

OUTPUT :

enter the limit : 4

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

COURSE OUTCOME 2(CO2)

PROGRAM NO : 10

DATE: 8/12/2021

AIM : Generate all factors of a number.

Program Source Code:

```
def factors(x):  
    print("the factor 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 : 4

the factor of 4 are:

1

2

4

COURSE OUTCOME 2(CO2)

PROGRAM NO : 10

DATE: 8/12/2021

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

Program Source Code:

```
import math

t_area = lambda b,h : 1/2*b*h
r_area = lambda l,b : l*b
s_area = lambda a : a*a

print("Area of Triangle :", t_area(10,20))
print("Area of Rectangle:", r_area(30,20))
print("Area of Square :", s_area(15))
```

OUTPUT :

Area of Triangle : 100.0

Area of Rectangle: 600

Area of Square : 225

COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

DATE:15/12/2021

AIM:Work with built-in packages

Time Module

```
import time
print("Current time in sec:",time.time())
print("Current time:",time.ctime())
print("Time After 30 sec:",time.ctime(time.time()+30))
t=time.localtime()
print("Time:",t)
print("Time-current year:",t.tm_year)
print("Time:-current month",t.tm_mon)
print("Time:-current day",t.tm_mday)
print("Time:-current hour",t.tm_hour)
print("Time:-current minute",t.tm_min)
print("Time:-current sec",t.tm_sec)
print("Time:-current week day",t.tm_wday)
print("Time:-current year day",t.tm_yday)
```

OUTPUT :

Current time in sec: 1639915265.630671

Current time: Sun Dec 19 17:31:05 2021

Time After 30 sec: Sun Dec 19 17:31:35 2021

```
Time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=19,  
tm_hour=17, tm_min=31, tm_sec=5, tm_wday=6, tm_yday=353,  
tm_isdst=0)
```

Time-current year: 2021

Time:-current month 12

Time:-current day 19

Time:-current hour 17

Time:-current minute 31

Time:-current sec 5

Time:-current week day 6

Time:-current year day 353

Math module

```
import math  
print(math.factorial(4))  
print (math.gcd(3, 6))  
print (math.sqrt(9))
```

OUTPUT :

24

3

3.0

Calendar module

```
import calendar  
mm=int(input("Enter month:"))  
yy=int(input("Enter year:"))  
print("\n")  
print(calendar.month(yy,mm))
```

OUTPUT

Enter month:1

Enter year:2022

January 2022

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						

DateTime module

```
import datetime  
t=datetime.time(22,56,20,67)  
print(t)  
print("Hour",t.hour)  
print("Minutes",t.minute)
```

```
print("Seconds",t.second)
print("Microsecond:",t.microsecond)
print("\n")
d=datetime.date.today()
print(d)
print("Year:",d.year)
print("Month:",d.month)
print("Day:",d.day)
d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2)
print(td)
d2=d1+td
print(d2)
dt=datetime.datetime.combine(d1,t)
print(dt)
```

OUTPUT:

22:56:20.000067

Hour 22

Minutes 56

Seconds 20

Microsecond: 67

2021-12-19

Year: 2021

Month: 12

Day: 19

2021-12-19

2 days, 0:00:00

2021-12-21

2021-12-19 22:56:20.000067

Statistics module

```
import statistics
print(statistics.mean([3,4,3]))
print(statistics.median([1, 3, 5, 7, 9, 11, 13]))
print(statistics.mode([1, 1, -3, 3, 7, -9]))
print(statistics.variance([1, 3, 5, 7, 9, 11]))
print(statistics.stdev([1, 3, 5, 7, 9, 11]))
```

OUTPUT:

3.3333333333333335

7

1

14

3.7416573867739413

Random module

```
import random
random.seed(10)
print(random.random())
mylist = ["apple", "banana", "cherry"]
```

```
print(random.sample(mylist, k=2))  
print(random.random())  
mylist2 = ["apple", "banana", "cherry"]  
random.shuffle(mylist2)  
print(mylist2)  
mylist3 = ["apple", "banana", "cherry"]  
print(random.choice(mylist3))
```

OUTPUT:

```
0.5714025946899135  
['banana', 'cherry']  
0.5780913011344704  
['cherry', 'banana', 'apple']  
banana
```

COURSE OUTCOME 3(CO3)

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)

- **graphics package**

circle module

```
def area(r):  
    return(3.14*r*r)
```

```
def perimeter(r):  
    return(2*3.14*r)
```

rectangle module

```
def area(l,b):  
    return(l*b)
```

```
def perimeter(l,b):  
    return(2*(l+b))
```

- **3dgraphics package**

sphere module

```
def area(r):  
    return(4*3.14*r*r)
```

```
def perimeter(r):  
    return(2*3.14*r)
```

cuboid module

```
def area(l,w,h):  
    return(2*l*w+2*l*h+2*h*w)
```

```
def perimeter(l,b,h):  
    return(4*(l+b+h))
```

```
from graphics import rectangle  
from graphics import circle  
from dgraphics import cuboid  
from dgraphics import sphere  
print("Rectangle:")  
l=int(input("Enter the length:"))  
b=int(input("Enter the breadth:"))  
print("Area=",rectangle.area(l,b))  
print("Perimeter=",rectangle.perimeter(l,b))  
print("\nCircle:")
```



```
r=int(input("Enter the radius:"))
print("Area=",circle.area(r))
print("Perimeter=",circle.perimeter(r))
print("\nCuboid:")
l=int(input("Enter the length:"))
w=int(input("Enter the width:"))
h=int(input("Enter the height:"))
b=int(input("Enter the breadth:"))
print("Area=",cuboid.area(l,w,h))
print("perimeter=",cuboid.perimeter(l,b,h))
print("\nSphere:")
r=int(input("Enter the radius:"))
print("Area=",sphere.area(r))
print("perimeter=",sphere.perimeter(r))
```

OUTPUT :

Rectangle:

Enter the length:2

Enter the breadth:2

Area= 4

Perimeter= 8

Circle:

Enter the radius:2

Area= 12.56

Perimeter= 12.56

Cuboid:

Enter the length:2

Enter the width:2

Enter the height:1

Enter the breadth:2

Area= 16

perimeter= 20

Sphere:

Enter the radius:2

Area= 50.24

perimeter= 12.56

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.

Program Source Code:

```
class rectangle:

    def __init__(self,length,breadth):

        self.length=length

        self.breadth=breadth


    def area(self):

        area=self.length*self.breadth

        print("Area=",area)

        return(area)


    def perimeter(self):

        per=2*(self.length+self.breadth)

        print("Perimeter=",per)


print("First Rectangle:")

b1=rectangle(2,2)

a1=b1.area()

b1.perimeter()
```

```
print("\nSecond Rectangle:")  
b2=rectangle(3,3)  
a2=b2.area()  
b2.perimeter()  
if a1 > a2:  
    print("\nArea of first rectangle is larger")  
else:  
    print("\nArea of second rectangle is larger")
```

OUTPUT :

First Rectangle:

Area= 4

Perimeter= 8

Second Rectangle:

Area= 9

Perimeter= 12

Area of second rectangle is larger

COURSE OUTCOME 4(CO4)

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.

Program Source Code:

```
class Bank:
    def __init__(self,bal=0):
        #self.accno=accno
        #self.name=name
        #self.acctype=acctype
        self.bal=bal
        name=input("Enter name:")
        print("Account for",name,"is created")

    def deposit(self):
        amount=int(input("Amount to deposit"))
        self.bal=self.bal+amount
        print("New balance:",self.bal)

    def withdarw(self):
        amount=int(input("Amount to withdraw"))
        if(self.bal>amount):
            self.bal=self.bal-amount
            print("New balance:",self.bal)
        else:
            print("insufficient amount")
            print("balance:",self.bal)

    def display(self):
        print("Current Balance:",self.bal)
        print("account")
        b1=Bank()
        opt='y'
        while(opt=='y'):
```

```
#print("your choice: 1. deposit \n 2. withdarw \n 3. display\n")
choice=int(input("your choice: 1. deposit \n 2. withdarw \n 3. display\n"))
if(choice == 1):
    b1.deposit()
elif(choice==2):
    b1.withdarw()
elif(choice==3):
    b1.display()
else:
    print("invalid")
opt=input("do you want to continue ('y'/'n')")
```

OUTPUT :

account

Enter name:akhil

Account for akhil is created

your choice: 1. deposit

2. withdarw

3. display

1

Amount to deposit5000

New balance: 5000

do you want to continue ('y'/'n')y

your choice: 1. deposit

2. withdarw

3. display

2

Amount to withdraw500

New balance: 4500

do you want to continue ('y'/'n')y

your choice: 1. deposit

2. withdarw

3. display

3

Current Balance: 4500

do you want to continue ('y'/'n')n

COURSE OUTCOME 4(CO4)

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.

Program Source Code:

```
class rectangle:
    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)

print("Enter the Details of Rectangle:1")
l1=int(input("Length:"))
w1=int(input("width:"))
r1=rectangle(l1,w1)
print("Enter the Details of Rectangle:2")
l2=int(input("Length:"))
w2=int(input("width:"))
r2=rectangle(l2,w2)
if(r1>r2):
    print("Rectangle 2 is larger!!")
else:
    print("Rectangle 1 is larger!!")
```


OUTPUT :

Enter the Details of Rectangle:1
Length:3
width:4

Enter the Details of Rectangle:2
Length:4
width:6

Rectangle 2 is larger!!

COURSE OUTCOME 4(CO4)

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.

Program Source Code:

```
class Time:
    def __init__(self,hour,minute,second):
        self.__hour=hour
        self.__minute=minute
        self.__second=second
    def __add__(self,a2):
        second=self.__second+a2.__second
        minute=self.__minute+a2.__minute
        hour=self.__hour+a2.__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 time1:")
h1=int(input("hour:"))
m1=int(input("minute:"))
s1=int(input("second"))
```

```
t1=Time(h1,m1,s1)
print("Enter time2:")
h2=int(input("hour:"))
m2=int(input("minute:"))
s2=int(input("second"))
t2=Time(h2,m2,s2)
hr,min,sec=t1+t2
print(hr,end=":")
print(min,end=":")
print(sec,end=" ")
```

OUTPUT :

Enter time1:

hour:2

minute:30

second23

Enter time2:

hour:1

minute:23

second45

3:54:8

COURSE OUTCOME 4(CO4)

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.

Program Source Code:

```
class publisher:
    def __init__(self,title,author):
        self.title=title
        self.author=author
    def display(self):
        print("Title:",self.title)
        print("Author:",self.author)

class book(publisher):
    def __init__(self,price,no_of_page):
        self.price=price
        self.no_of_page=no_of_page
    def display(self):
        print("Price:",self.price)
        print("No. of Pages:",self.no_of_page)

class python(book):
    def __init__(self,title,author,price,no_of_page):
        publisher.__init__(self,title,author)
        book.__init__(self,price,no_of_page)
    def display(self):
        print("Title:",self.title)
        print("Author:",self.author)
        print("Price:",self.price)
        print("No. of Pages:",self.no_of_page)
p=python("Python Programming","AnilKumar",1000,120)
p.display()
```

OUTPUT :

Title: Python Programming

Author: AnilKumar

Price: 1000

No. of Pages: 120

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.

Program Source Code:

```
f1=open("sample.txt","w")

f1.write("This is my first line.\n This is my second line \n This is my third line")

f1=open("sample.txt","r")

ff=f1.readlines()

print(ff)
```

OUTPUT :

```
['This is my first line.\n', ' This is my second line \n', ' This is my third line']
```

This is my first line.

This is my second line

This is my third line

COURSE OUTCOME 5(CO5)

PROGRAM NO: 2

DATE:30/1/2022

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

Program Source Code:

```
f1=open("sample.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 :

This is my second line

COURSE OUTCOME 5(CO5)**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.

Program Source Code:

```
import csv

with open('departments.csv', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=' ', quotechar='"')
    for r in data:
        print(' '.join(r))
```

department.csv

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```

OUTPUT :

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```


COURSE OUTCOME 5(CO5)

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

Program Source Code:

```
import csv

with open('departments.csv', newline='') as csvfile:
    data = csv.DictReader(csvfile)
    print("id Department")
    print("-----")
    for r in data:
        print(r['department_id'], " ", r['department_name']
```

department.csv

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```

OUTPUT :

```
id Department
-----
10 Administration
20 Marketing
30 Purchasing
```

COURSE OUTCOME 5(CO5)

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.

Program Source Code:

```
import csv

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

cars = [
    {'No': 1, 'Company': 'Ferrari', 'Model': '488 GTB'},
    {'No': 2, 'Company': 'Porsche', 'Model': '918 Spyder'},
    {'No': 3, 'Company': 'Bugatti', 'Model': 'La Voiture Noire'},
]

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

with open('cars.csv', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=' ', quotechar='"')
    for r in data:
        print(' '.join(r))
```

cars.csv

```
No,Company,Model
1,Ferrari,488 GTB
2,Porsche,918 Spyder
3,Bugatti,La Voiture Noire
```

OUTPUT

No,Company,Model

1,Ferrari,488, GTB

2,Porsche,918, Spyder

3,Bugatti,La, Voiture, Noire