SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-2022

20 MCA 132 PROGRAMMING LABORATORY RECORD

Submitted by

SITHARA MOL K S

REG NO: SNG21MCA-2035

In partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATIONS

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682311

(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 **SITHARA MOL K S***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 2021-2022.*

Kadayiruppu Course Instructor

Date:…….

Head of the Department

Prof. Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2035 on……

External Examiner Internal Examiner

SL NO.	DATE	NAME OF EXPERIMENT	PAGE NO.	REMARK
I	CO 1			
1.	03-11-2021	Familiarizing Text Editor, IDE, Code Analysis Tools etc.	1	
2.	08-11-2021	Find Leap Year.	2	
3.a)	10-11-2021	Generate positive list of numbers from a given list of integers.	3	
3.b)	10-11-2021	Find the Square of N number.	3	
3.c)	10-11-2021	Form a list of vowels selected from a given word.	4	
3.d)	10-11-2021	List ordinal value of a word.	4	
4.	15-11-2021	Count the occurrences of each words.	5	
5.	15-11-2021	Prompt the user for a list of integers.	6	
6.	17-11-2021	Count the occurrences of ‘a’ in list	7	
7.	17-11-2021	Checking lists are of same length, sums to same value, whether any value occurs in both.	8	
8.	22-11-2021	Get a string from an input string and replace a character.	10	
9.	22-11-2021	Create a string from given string where first and last characters exchanged.	11	
10.	24-11-2021	Accept the radius from user and find area of circle. Create a list of colors.	12	
11.	24-11-2021	Find biggest of 3 numbers.	13	
12.	24-11-2021	Print extension of files.	14	
13	29-11-2021	Create a list of colors. Display first and last colors.	15	
14.	29-11-2021	Accept an integer n and computer n+nn+nnn.	16	
15.	29-11-2021	Print out all colors from color-list1 not contained in color-list2.	17	
16.	29-11-2021	Create a single string separated with space from two strings by swapping the character at position.	18	

COURSE OUTCOME 1 (CO 1)

CO 1:PROGRAM 1 **DATE: 03-11-2021**

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

It is a Graphical User Interface (GUI) where programmers write their code and produce the final products. An IDE basically unifies all essential tools required for software development and testing, which in turn helps the programming maximize his output.

Features of IDE:-

1. Code Editor

2. Syntax Highlighting

3. Auto completion code

4. Debugger

5. Compiler

6. Language Support

IDLE is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- coded in 100% pure Python, using the tkinter GUI toolkit.
- cross-platform: works mostly the same on Windows, Unix, and macOS.
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages.
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features.
- search within any window, replace within editor windows, and search through multiple files (grep).
- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces.
- configuration, browsers, and other dialogs

CO 1:PROGRAM2 **DATE: 08-11-2021**

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

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

OUTPUT:

```
enter start year:2000
enter end year:2010
leap year are:2004 2008
```

CO 1: PROGRAM 3 **DATE: 10-11-2021**

List comprehensions:

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

```
for i in [-19,20,90,-56,87]:
  if i>=0:
    print(i)
```

OUTPUT:

20

90

87

AIM: b) Square of N number

```
n=int(input("enter the limit:"))
for i in range(1,n+1):
    s=i*i
    print(s)
```

OUTPUT:

enter the limit:5

1

4

9

16

25

AIM: c) Form a list of vowels selected from a given word

```
c=str(input("enter a word:"))
print("orginal word:",c)
print("vowels are")
for i in c:
   if i in "aeiouAEIOU":
      print([i])
```

OUTPUT:

enter a word:sithara

orginal word: sithara

vowels are:

['i']

['a']

['a']

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

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

OUTPUT

Enter a word:sithara

Ordinal values corresponding to each element is:

s:115 i:105 t:116 h:104 a:97 r:114 a:97

CO 1:PROGRAM 4 **DATE: 15-11-2021**

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 string : sithara
count of the occurrence:[('sithara', 1)]
```

CO 1:PROGRAM 5 **DATE: 15-11-2021**

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 limit:3
Enter {s} values
56
90
32

The list after assinging:

56
90
32
```

CO 1:PROGRAM6 **DATE: 17-11-2021**

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

```
lst = ["a","b","c","a"]
occ = lst.count("a")
print("Occurrences of 'a' :",occ)
```

OUTPUT

```
Occurrences of 'a' : 2
```

CO 1:PROGRAM 7 **DATE: 17-11-2021**

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]

CO 1:PROGRAM 8 **DATE: 22-11-2021**

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

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

OUTPUT:

malayala$

CO 1:PROGRAM 9 **DATE: 22-11-2021**

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

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

OUTPUT:

```
enter a string:python
nythop
```

CO 1:PROGRAM 10 **DATE: 24-11-2021**

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

```
r=float(input("Enter the radius:"))
area=3.14*r*r
print("area=",area)
```

OUTPUT:

Enter the radius2

area= 12.56

CO1:PROGRAM 11 **DATE: 24-11-2021**

AIM: Find biggest of 3 numbers entered

```
a=int(input("enter 1st no:"))
b=int(input("enter 2nd no:"))
c=int(input("enter 3rd no:"))
if(a>b and a>c):
 print(a,"is the largest")
elif(b>c):
 print(b,"is the largest")
elif(c>a):
 print(c,"is the largest")
```

OUTPUT:

```
enter 1st no:56
enter 2nd no:12
enter 3rd no:4589
4589 is the largest
```

CO 1:PROGRAM 12 **DATE: 24-11-2021**

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

```
n1=input("Enter file name:")
f=n1.split(".")
print("extension of file is:"+f[-1])
```

OUTPUT:

Enter file name:python.java

extension of file is: java

CO 1:PROGRAM 13 **DATE: 29-11-2021**

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

```
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 color:red
Enter the color:blue
Enter the color:yellow
['red', 'blue', 'yellow']
red
yellow
```

CO 1:PROGRAM 14 **DATE: 29-11-2021**

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

```
n=input("enter a number:")
a=int("%s" %n)
b=int("%s%s" %(n,n))
c=int("%s%s%s" %(n,n,n))
print("n + nn + nnn:",a+b+c)
```

OUTPUT:

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

CO 1:PROGRAM 15 **DATE: 29-11-2021**

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

```
l1=set(["red","blue","green","white"])
l2=set(["red","blue","green","orange"])
print(l1.difference(l2))
```

OUTPUT:

{'white'}

CO 1:PROGRAM 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"
q=a[0]
p=b[0]
c=b[0]+a[1:len(a)]+ " " +a[0]+b[1:len(b)]
print(c)
```

OUTPUT:

jython pava

CO 1:PROGRAM 17 **DATE: 01-12-2021**

AIM: Sort dictionary in ascending and descending order.

```
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print("orginal list",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:

```
orginal list {7: 2, 3: 4, 9: 3, 2: 12, 0: 0}
Dictionary in ascending order by value  [(0, 0), (7, 2), (9, 3), (3, 4), (2, 12)]
Dictionary in descending order by value : {2: 12, 3: 4, 9: 3, 7: 2, 0: 0}
```

CO 1:PROGRAM 18 **DATE: 01-12-2021**

AIM: Merge two dictionaries

```
m={'p':200,'q':600}
n={'l':500,'s':100}
print("dictionary 1:",m)
print("dictionary 2:",n)
m1=m.copy()
m1.update(n)
print("merged dictionary:",m1)
```

OUTPUT:

dictionary 1: {'p': 200, 'q': 600}

dictionary 2: {'l': 500, 's': 100}

merged dictionary: {'p': 200, 'q': 600, 'l': 500, 's': 100}

CO 1:PROGRAM 19 **DATE: 01-12-2021**

AIM: Find gcd of 2 numbers.

```
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: 10

Enter 2nd number: 8

GCD : 2

CO 1:PROGRAM 20 **DATE: 01-12-2021**

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 numbers:",num)
```

OUTPUT:

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

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

COURSE OUTCOME 2 (CO 2)

CO 2:PROGRAM 1 **DATE: 06-12-2021**

AIM: Program to find the factorial of a number

```
n=int(input("enter a no:"))
f=1
for i in range(1,n+1):
 f=i*f
print("factotial=",f)
```

OUTPUT:

```
enter a no:5
factotial= 120
```

CO 2:PROGRAM 2 **DATE: 06-12-2021**

AIM: Generate Fibonacci series of N terms

```
n=int(input("enter the limit"))
x=0
y=1
s=0
print("fibonancci series:")
for i in range(1,n):
  x=y
  y=s
  s=x+y
  print(s)
```

OUTPUT:

```
enter the limit5
fibonancci series:
1
1
2
3
```

CO 2:PROGRAM 3 **DATE: 06-12-2021**

AIM: Find the sum of all items in a list

```
n = [1,2,4,7,9]
total = sum(n)
print("Sum of list : ",total)
```

OUTPUT:

```
Sum of list :  23
```

CO 2:PROGRAM 4 **DATE: 06-12-2021**

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

CO 2:PROGRAM 5 **DATE: 06-12-2021**

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

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

OUTPUT:

```
Enter the number of rows: 3
1
2 4
3 6 9
```

CO 2:PROGRAM 6 **DATE: 06-12-2021**

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

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

OUTPUT:

Enter the string : sithara

Count of all characters : {'s': 1, 'i': 1, 't': 1, 'h': 1, 'a': 2, 'r': 1}

CO 2:PROGRAM 7 **DATE: 08-12-2021**

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("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
```

CO 2:PROGRAM 8 **DATE: 08-12-2021**

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 list:3
Enter element 1here
Enter element 2is
Enter element 3python
Longest Word: python
Length of longest word : 6
```

CO 2:PROGRAM 9 **DATE: 08-12-2021**

AIM: Construct following pattern using nested loop

```
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:3

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

CO 2:PROGRAM 10 **DATE: 08-12-2021**

AIM: Generate all factors of a number. def print_factors

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

OUTPUT:

```
enter a number16
the factor of 16
1
2
4
8
16
```

CO 2:PROGRAM 11 **DATE: 08-12-2021**

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

```
import math
Tri_area=lambda b,h:1/2*b*h
Rect_area=lambda l,b:l*b
Squa_area=lambda l:l*l
print("area of triangle",Tri_area(2,5))
print("area of rectangle",Rect_area(6,10))
print("area of square",Squa_area(30))
```

OUTPUT:

```
area of triangle 5.0
area of rectangle 60
area of square 900
```

COURSE OUTCOME 3 (CO 3)

CO 3:PROGRAM 1 **DATE: 13-12-2021**

AIM: Work with built-in packages

A)Statistics.py

```
import statistics

# Calculate average values
print("Mean : ",statistics.mean([1, 3, 5, 7, 9, 11, 13]))
print("Mean : ",statistics.mean([1, 3, 5, 7, 9, 11]))
print("Mean : ",statistics.mean([-11, 5.5, -3.4, 7.1, -9, 22]))

print("==============================")

# Calculate middle values
print("Median : ",statistics.median([1, 3, 5, 7, 9, 11, 13]))
print("Median : ",statistics.median([1, 3, 5, 7, 9, 11]))
print("Median : ",statistics.median([-11, 5.5, -3.4, 7.1, -9, 22]))


print("==============================")

# Calculate the mode
print("Mode :",statistics.mode([1, 3, 3, 3, 5, 7, 9, 11]))
print("Mode :",statistics.mode([1, 1, 3, -5, 7, -9, 11]))
print("Mode :",statistics.mode(['red', 'green', 'blue', 'red']))

print("==============================")

# Calculate the variance from a sample of data
print("Varience :",([1, 3, 5, 7, 9, 11]))
print("Varience :",statistics.variance([2, 2.5, 1.25, 3.1, 1.75, 2.8]))
print("Varience :",statistics.variance([-11, 5.5, -3.4, 7.1]))
print("Varience :",statistics.variance([1, 30, 50, 100]))

print("==============================")

# Calculate harmonic mean
print("Hermonic mean",statistics.harmonic_mean([40, 60, 80]))
print("Hermonic mean",statistics.harmonic_mean([10, 30, 50, 70, 90]))

print("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-")
```

OUTPUT:

```
Mean :  7
Mean :  6
Mean :  1.8666666666666667
==============================
Median :  7
Median :  6.0
Median :  1.05
==============================
Mode : 3
Mode : 1
Mode : red
==============================
Varience : [1, 3, 5, 7, 9, 11]
Varience : 0.4796666666666667
Varience : 70.80333333333334
Varience : 1736.9166666666667
==============================
Hermonic mean 55.38461538461538
Hermonic mean 27.97513321492007
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

B)random.py

```
import random

print(random.random())

print("==============================")

mylist = ["apple", "banana", "cherry"]
random.shuffle(mylist)
print(mylist)

print("==============================")

random.seed(10)
print(random.random())

print("==============================")

mylist = ["apple", "banana", "cherry"]
print(random.choice(mylist))

print("==============================")

print(random.randrange(3, 9))
```

OUTPUT:

```
0.25174248185744386
==============================
['apple', 'cherry', 'banana']
==============================
0.5714025946899135
==============================
banana
==============================
6
```

C)Math.py

```
import math
print("value of pi is",math.pi)

import math as m
print("value of pi from m",m.pi)

from math import pi,sqrt
print("value of pi",pi)
print("square root of 4 is:",sqrt(4))
print(math.cos(90))
print(math.sin(60))
print(math.tan(60))
print(math.tan(45))
```

OUTPUT:

```
value of pi is 3.141592653589793
value of pi from m 3.141592653589793
value of pi 3.141592653589793
square root of 4 is: 2.0
-0.4480736161291701
-0.3048106211022167
0.320040389379563
1.6197751905438615
```

D)Datetime.py

```
import datetime

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

print("==============================")


d = datetime.date.today()
```

```
print(d)
td = datetime.timedelta(days=2)
print(td)

d2 = d+td
print("After adding two days :",d2)
print("d2-d",d2-d)
print("d2>d",d2>d)

d1 = datetime.date.today()
t1 = datetime.time(12,44,56)
print("Date and Time : ",d1, t1)
```

OUTPUT:

```
22:56:44
Hour :  22
Minute :  56
Second :  44
==============================
2021-12-22
2 days, 0:00:00
After adding two days : 2021-12-24
d2-d 2 days, 0:00:00
d2>d True
Date and Time :  2021-12-22 12:44:56
```

E)calendar.py

```
import calendar

mm = int(input("enter  month:"))

yy = int(input("enter year:"))

print(calendar.month(yy,mm))

print(calendar.calendar(2021))
```

OUTPUT:

```
enter  month:2
enter year:2002
    February 2002
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
```

```
                                  2021

      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        1  2  3  4  5  6  7        1  2  3  4  5  6  7
 4  5  6  7  8  9 10        8  9 10 11 12 13 14        8  9 10 11 12 13 14
11 12 13 14 15 16 17       15 16 17 18 19 20 21       15 16 17 18 19 20 21
18 19 20 21 22 23 24       22 23 24 25 26 27 28       22 23 24 25 26 27 28
25 26 27 28 29 30 31                                  29 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                       1  2           1  2  3  4  5  6
 5  6  7  8  9 10 11        3  4  5  6  7  8  9        7  8  9 10 11 12 13
12 13 14 15 16 17 18       10 11 12 13 14 15 16       14 15 16 17 18 19 20
19 20 21 22 23 24 25       17 18 19 20 21 22 23       21 22 23 24 25 26 27
26 27 28 29 30             24 25 26 27 28 29 30       28 29 30
                           31

      July                       August                     September
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  2  3  4  5
 5  6  7  8  9 10 11        2  3  4  5  6  7  8        6  7  8  9 10 11 12
12 13 14 15 16 17 18        9 10 11 12 13 14 15       13 14 15 16 17 18 19
19 20 21 22 23 24 25       16 17 18 19 20 21 22       20 21 22 23 24 25 26
26 27 28 29 30 31          23 24 25 26 27 28 29       27 28 29 30
                           30 31

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

F)Time.py

```
import time
print("current time in sec:",time.time())
print("current time:",time.ctime())
print ("current time after 30 se:",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 weekend day no:",t.tm_wday)
print("day of year:",t.tm_yday)
```

OUTPUT:

current time in sec: 1640165761.415304

current time: Wed Dec 22 15:06:01 2021

current time after 30 se: Wed Dec 22 15:06:31 2021

time time.struct_time(tm_year=2021, tm_mon=12, tm_mday=22, tm_hour=15, tm_min=6, tm_sec=1, tm_wday=2, tm_yday=356, tm_isdst=0)

current year: 2021

current month: 12

current day: 22

current hour: 15

current weekend day no: 2

day of year: 356

CO 3:PROGRAM 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)

rectangle.py

```
def rectangle(l,b):
    print("Area of rectangle :",l*b)
    print("Perimeter of rectangle :",2*(l+b))
```

circle.py

```
def circle(r)
    print("Area of Circle :",3.14*r*r)
    print("Perimeter of rectangle :",2*3.14*r)
```

cuboid.py

```
def cuboid(l,b,h):
    print("Area of Cuboid: ",(2*l*b)+(2*l*h)+(2*h*b))
    print("Perimeter of Cuboid: ", 4*(l+b+h))
```

sphere.py

```
def sphere(r):
  print("area of sphere=",4*3.14*r*r)
  print("perimeter of sphere =",2*3.14*r)
```

appackage.py

```
from graphics import circle
from graphics import rectangle
from graphics import sphere
from graphics import cuboid

r=int(input("enter the radius:"))
circle.circles(r)

l=int(input("enter the lenght:"))
b=int(input("enter the bredth:"))
rectangle.rectangles(l,b)

r=int(input("enter the radius:"))
sphere.sphere(r)
```

```
l=int(input("enter the lenght:"))
b=int(input("enter the bredth:"))
h=int(input("enter the height:"))
cuboid.cuboid(l,b,h)
```

OUTPUT:

```
enter the radius:3
area of circle= 28.259999999999998
perimeter of circle= 18.84
enter the lenght:2
enter the bredth:8
area of rec= 16
perimeter of rec= 20
enter the radius:9
area of sphere= 1017.36
perimeter of sphere = 56.52
enter the lenght:1
enter the bredth:2
enter the height:3
Area of Cuboid:  22
Perimeter of Cuboid:  24
```

COURSE OUTCOME 4 (CO 4)

CO 4:PROGRAM 1 **DATE: 03-01-2022**

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: 2
Enter breadth of rectangle: 7
Area of rectangle: 14
perimeter of rectangle: 11
obj1 is greater
```

CO 4:PROGRAM 2 **DATE: 05-01-2022**

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:

```
enter the name of account holder:sithara
enter the account no:1000011
The account is created

 name of account: sithara

 account no: 1000011
 enter the amount:4000
Enter amount to be Withdrawn:1500

You Withdraw: 1500.0

Available Balance = 2500.0
```

CO 4:PROGRAM 3 **DATE: 05-01-2022**

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

```
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)
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:3
width  1 rectangle:4
Length 2 rectangle:7
width 2 rectangle:9
Rectangle 2 is larger!!
```

CO 4:PROGRAM 4 **DATE: 10-01-2022**

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:2
 enter the second:4
Enter  2 time:
enter the hour:7
enter the minute:10
enter the second:5
12:12:9
```

CO 4:PROGRAM 5 **DATE: 10-01-2022**

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 __init__(self,pname):
    self.pname=pname


  def display(self):
   print("Publisher Name:",self.pname)


class book(publisher):


  def get(self,title,author):
    self.title=title
    self.author=author


  def display(self):
   print("Title Name:",self.title)
   print("Author Name:",self.author)


class python(book):


 def __init__(self,price,nop,pname):
  super().__init__(pname)
  self.price=price
  self.nop=nop


 def details(self):
  print("Price:",self.price)
  print("No of pages:",self.nop)
```

```
s1=python(1200,192,"KIRAN")
s1.get("PYTHON PROGRAMMING","KIRAN")
s1.display()
s1.details()
```

OUTPUT:

Title Name: PYTHON PROGRAMMING

Author Name: KIRAN

Price: 1200

No of pages: 192

COURSE OUTCOME 5(CO 5)

CO 5:PROGRAM 1 **DATE: 17-01-2022**

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

```
f1=open("firstfile.txt","w")
f1.write("This is my first file in python.\nWant to work with files\nThis is my third line")
f1.close()
f1=open("firstfile.txt","r")
print("Name of file:",f1.name)
print("Close of file:",f1.close)
print("Mode of file:",f1.mode)
print(f1.read())
print("------------------")
f1.seek(0,0)
print(f1.read(15))
print(f1.readline())
print(f1.readline())
f1.seek(0,0)
ff=f1.readlines()
print(ff)
print("No of lines : ",len(ff))
print("------------------")
#import os
#os.rename("firstfile.txt","secfile.txt")
#print(f1.name)
```

```
#f1.close()
```

OUTPUT:

```
Name of file: firstfile.txt
Close of file: <built-in method close of _io.TextIOWrapper object at 0x000002895B956260>
Mode of file: r
This is my first file in python.
Want to work with files
This is my third line
----------
This is my firs
t file in python.
Want to work with files

['This is my first file in python.\n', 'Want to work with files\n', 'This is my third line']
No of lines :  3
----------
```

CO 5:PROGRAM 2 **DATE: 17-01-2022**

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

```
f1 = open("firstfile.txt","r")
for x in f1:
   print(x)
print("----------")
f1.seek(0,0)
ff=f1.readlines()
print("Looping through the file using Readline")
for x in range(0,len(ff)):
   if(x%2==0):
      print(ff[x])
print("----------")

f2= open("Even.txt","w")
f2.write(ff[x])
```

OUTPUT:

This is my first file in python.

Want to work with files

This is my third line

Looping through the file using Readline

This is my first file in python.

This is my third line

CO 5:PROGRAM 3 **DATE: 31-01-2022**

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

```
import csv
 # csv file name
filename = "username.csv"
 # initializing the titles and rows list
fields = []
rows = []
 # reading csv file
cf=open(filename, 'r')
 # creating a csv reader object
csvreader = csv.reader(cf)
 # extracting field names through first row
fields = next(cf)
print(fields)
 # extracting each data row one by one
for r in csvreader:
 rows.append(r)
 #print the list containing the rows of csv file
print(rows)
print(".....")
print('\nFirst 3 rows are:\n')
for r in rows[:3]:
 print(*r)
 print()
print("The file content")
for sl in rows:
 for l in sl:
  print(l),
 #print(l,end=" ")
  print()
cf.close()
```

OUTPUT:

Username; Identifier;Firstname;Lastname

[['booker12;9012;Rachel;Booker'], ['grey07;2070;Laura;Grey'], ['johnson81;4081;Craig;Johnson'], ['jenkins46;9346;Mary;Jenkins'], ['smith79;5079;Jamie;Smith']]

.....

First 3 rows are:

booker12;9012;Rachel;Booker

grey07;2070;Laura;Grey

johnson81;4081;Craig;Johnson

The file content

booker12;9012;Rachel;Booker

grey07;2070;Laura;Grey

johnson81;4081;Craig;Johnson

jenkins46;9346;Mary;Jenkins

smith79;5079;Jamie;Smith

CO 5:PROGRAM 4 **DATE: 31-01-2022**

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

```
import csv
filename = "names.csv"
cf=open(filename, 'r')
 #csvreader = csv.reader(cf)
data = csv.DictReader(cf)
print("No Company")
for r in data:
 print(r['No'], r['Company'])
```

OUTPUT:

No Company

1 Ferrari

2 Porsche

3 Bugatti

4 Rolls Royce

5 BMW

CO 5:PROGRAM 5 **DATE: 31-01-2022**

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)
#print(".....")
filename = "names1.csv"
cf=open(filename, 'r')
rows=[]
csvreader = csv.reader(cf)
for r in csvreader:
 rows.append(r)
for r in rows[:3]:
 print(*r)
```

OUTPUT:

```
No Company Car Model

1 Ferrari 488 GTB
```