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

AMALESH C V

REG NO: SNG21MCA-2006

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 ***AMALESH C V*** *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-2006 on ------------

External Examiner InternalExaminer

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.

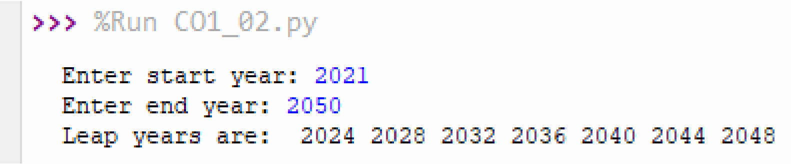
PROGRAM NO: 2

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 CO1_02.py

  Enter start year: 2021
  Enter end year: 2050
  Leap years are:  2024 2028 2032 2036 2040 2044 2048
```

PROGRAM NO: 3

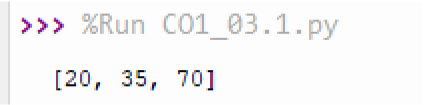
DATE:24/11/2021

AIM: List comprehensions:

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

```
list =[-10,20,35,-67,70]
for i in list:
      if(i>0):
            print(i)
```

OUTPUT

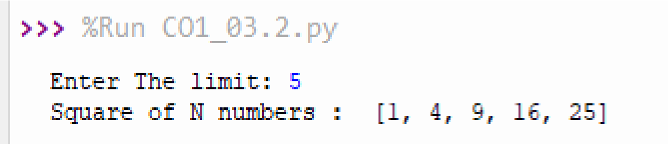


```
>>> %Run CO1_03.1.py

  [20, 35, 70]
```

- Square of N number

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

OUTPUT

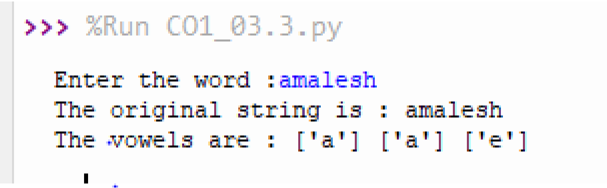


```
>>> %Run CO1_03.2.py

  Enter The limit: 5
  Square of N numbers :  [1, 4, 9, 16, 25]
```

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

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

OUTPUT

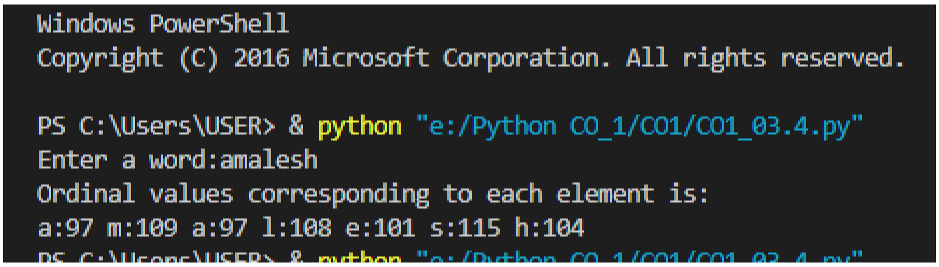


```
>>> %Run CO1_03.3.py

  Enter the word :amalesh
  The original string is : amalesh
  The vowels are : ['a'] ['a'] ['e']
```

- **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



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\USER> & python "e:/Python CO_1/CO1/CO1_03.4.py"
Enter a word:amalesh
Ordinal values corresponding to each element is:
a:97 m:109 a:97 l:108 e:101 s:115 h:104
PS C:\Users\USER> & python "e:/Python CO_1/CO1/CO1_03.4.py"
```

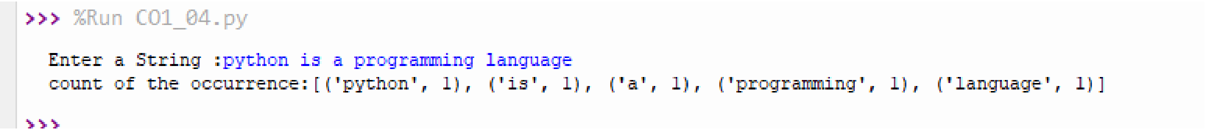
PROGRAM NO: 4

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



```
>>> %Run CO1_04.py

  Enter a String :python is a programming language
  count of the occurrence:[('python', 1), ('is', 1), ('a', 1), ('programming', 1), ('language', 1)]

>>>
```

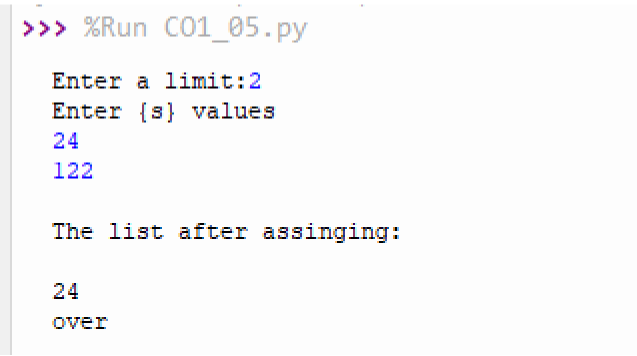
PROGRAM NO: 5

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



```
>>> %Run CO1_05.py

  Enter a limit:2
  Enter {s} values
  24
  122

  The list after assinging:

  24
  over
```

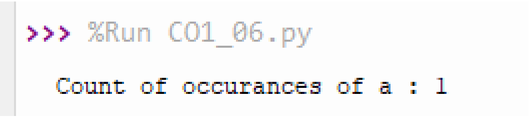
PROGRAM NO: 6

DATE:24/11/2021

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



```
>>> %Run CO1_06.py
  Count of occurances of a : 1
```

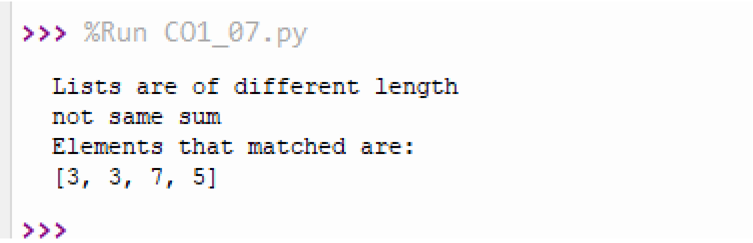
PROGRAM NO: 7

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

```
lst1=[12,3,4,3,6,7,9,11,23,5]
lst2=[32,3,35,7,5,20,65,1]
s=int(0)
c=int(0)

if len(lst1)==len(lst2):
   print("Lists are of same length")
else:
   print("Lists are of different length")

for i in range(0,len(lst1) and len(lst2)):
      s = lst1[i]
      c = c+lst2[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(lst2)):
      if lst1[i]==lst2[j]:
            l.append(lst1[i] and lst2[j])
      else:
         continue
print(l)
```

OUTPUT



```
>>> %Run CO1_07.py

  Lists are of different length
  not same sum
  Elements that matched are:
  [3, 3, 7, 5]

>>>
```

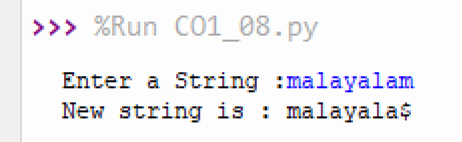
PROGRAM NO: 8

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]

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

OUTPUT



```
>>> %Run CO1_08.py

  Enter a String :malayalam
  New string is : malayala$
```

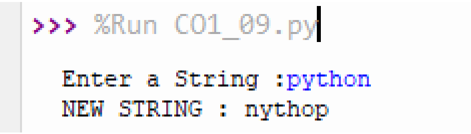
PROGRAM NO: 9

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 :")
newstr = str[-1:] + str[1:-1] + str[:1]
print("New string :",newstr)
```

OUTPUT



```
>>> %Run CO1_09.py

  Enter a String :python
  NEW STRING : nythop
```

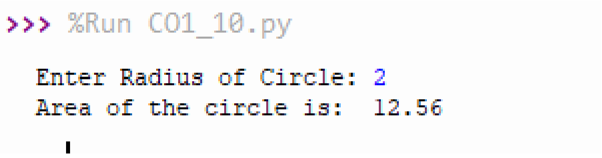
PROGRAM NO: 10

DATE:24/11/2021

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

```
pi = 3.14
r = float(input("Enter the radius of circle :"))
area = pi*r**2
print("Area of circle :", area)
```

OUTPUT



```
>>> %Run CO1_10.py

  Enter Radius of Circle: 2
  Area of the circle is:  12.56
```

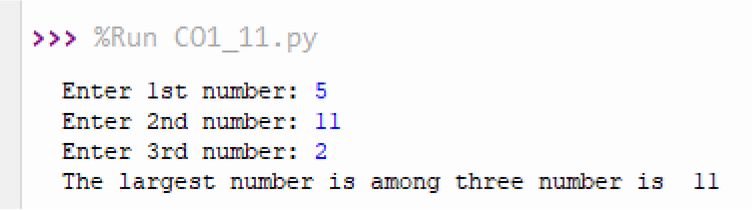
PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

```
a = int(input("Enter First No:"))
b = int(input("Enter Second No:"))
c = int(input("Enter Third No:"))

if(a > b and a>c):
      print(a,"is largest")
elif(b > c):
      print(b,"is largest")

elif(c > a):
      print(c,"is largest")
```

OUTPUT



```
>>> %Run CO1_11.py

  Enter 1st number: 5
  Enter 2nd number: 11
  Enter 3rd number: 2
  The largest number is among three number is  11
```

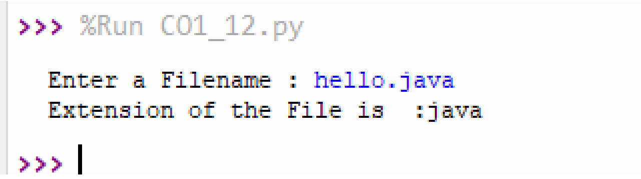
PROGRAM NO: 12

DATE:29/11/2021

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

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

OUTPUT



```
>>> %Run CO1_12.py

  Enter a Filename : hello.java
  Extension of the File is  :java

>>> 
```

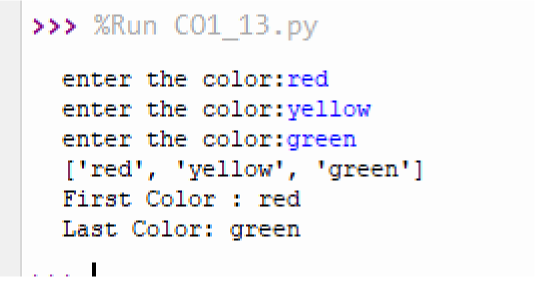
PROGRAM NO: 13

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("First Color :",a[0])
print("Last Color:",a[2])
```

OUTPUT



```
>>> %Run CO1_13.py

  enter the color:red
  enter the color:yellow
  enter the color:green
  ['red', 'yellow', 'green']
  First Color : red
  Last Color: green
```

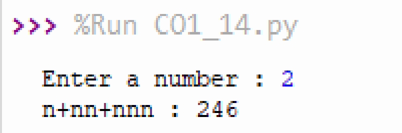
PROGRAM NO: 14

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



```
>>> %Run CO1_14.py

  Enter a number : 2
  n+nn+nnn : 246
```

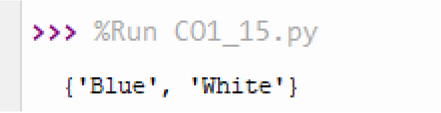
PROGRAM NO: 15

DATE:29/11/2021

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

```
lst1 = set(["White", "Pink", "Red", "Blue"])
lst2 = set(["Red", "Green", "Pink"])
print(lst1.difference(lst2))
```

OUTPUT



```
>>> %Run CO1_15.py
  {'Blue', 'White'}
```

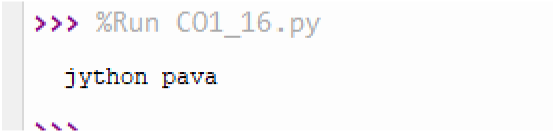
PROGRAM NO: 16

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



```
>>> %Run CO1_16.py
  jython pava
>>>
```

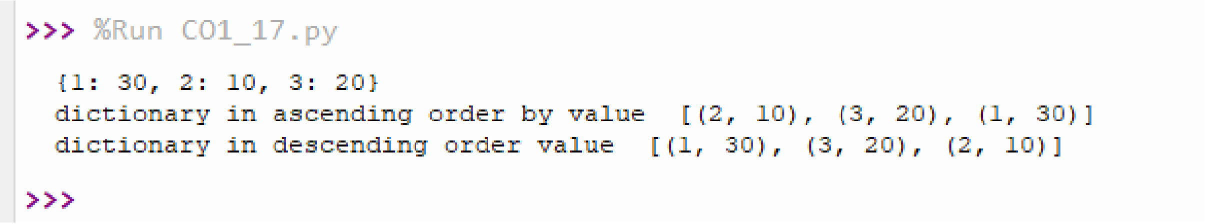
PROGRAM NO: 17

DATE:29/11/2021

AIM: Sort dictionary in ascending and descending order.

```
import operator
d={1:30,2:10,3:20}
print(d)
sort=sorted(d.items(),key=operator.itemgetter(1))
print("dictionary in ascending order by value ",sort)
sort2=sorted(d.items(),key=operator.itemgetter(1),reverse=True)
print("dictionary in descending order value ",sort2)
```

OUTPUT



```
>>> %Run CO1_17.py

  {1: 30, 2: 10, 3: 20}
  dictionary in ascending order by value  [(2, 10), (3, 20), (1, 30)]
  dictionary in descending order value  [(1, 30), (3, 20), (2, 10)]

>>>
```

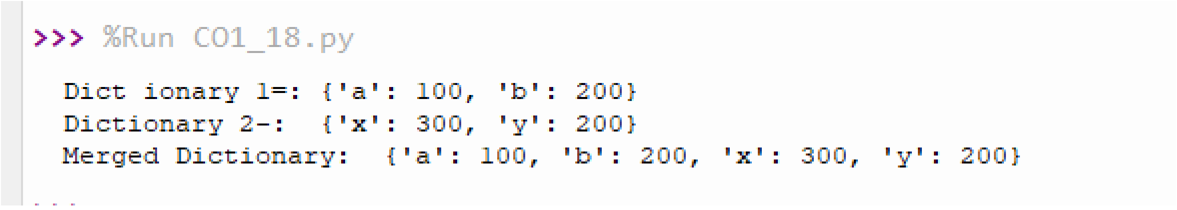
PROGRAM NO: 18

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



```
>>> %Run CO1_18.py

  Dict ionary 1=: {'a': 100, 'b': 200}
  Dictionary 2-:  {'x': 300, 'y': 200}
  Merged Dictionary:  {'a': 100, 'b': 200, 'x': 300, 'y': 200}
...
```

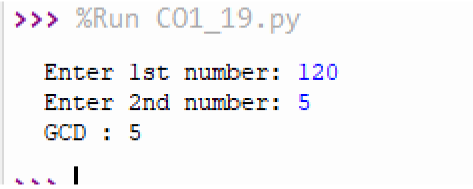
PROGRAM NO: 19

DATE:29/11/2021

AIM: Find gcd of 2 numbers.

```
x= int(input("Enter 1st 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



```
>>> %Run CO1_19.py

  Enter 1st number: 120
  Enter 2nd number: 5
  GCD : 5

>>>
```

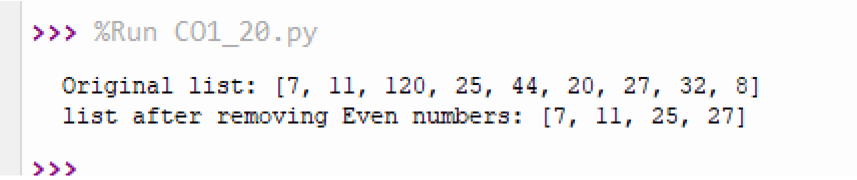
PROGRAM NO: 20

DATE:29/11/2021

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

```
num = [1,2,3,4,5,6,7,8,9,10]
print( "Original list:",num)
num = [x for x in num if x%2!=0]
print("list after removing Even numbers:",num)
```

OUTPUT



```
>>> %Run CO1_20.py

  Original list: [7, 11, 120, 25, 44, 20, 27, 32, 8]
  list after removing Even numbers: [7, 11, 25, 27]

>>>
```

II COURSE OUTCOME 2(CO2)

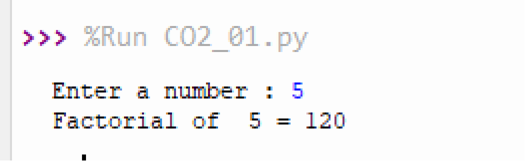
PROGRAM NO: 1

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



```
>>> %Run CO2_01.py

  Enter a number : 5
  Factorial of  5 = 120
```

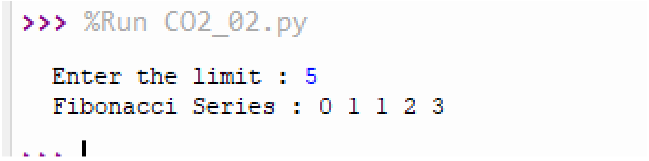
PROGRAM NO: 2

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



```
>>> %Run CO2_02.py

  Enter the limit : 5
  Fibonacci Series : 0 1 1 2 3

...
```

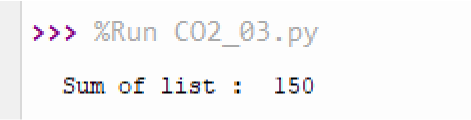
PROGRAM NO: 3

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



```
>>> %Run CO2_03.py

  Sum of list :  150
```

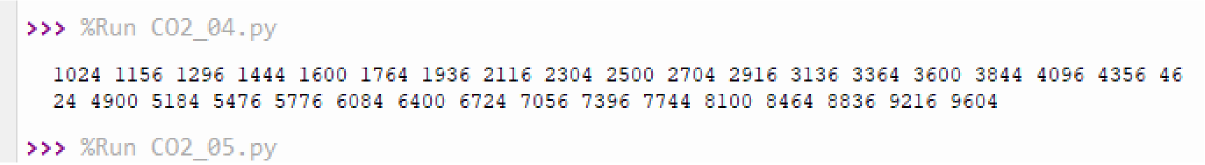
PROGRAM NO: 4

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



```
>>> %Run CO2_04.py

  1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 3136 3364 3600 3844 4096 4356 46
  24 4900 5184 5476 5776 6084 6400 6724 7056 7396 7744 8100 8464 8836 9216 9604

>>> %Run CO2_05.py
```

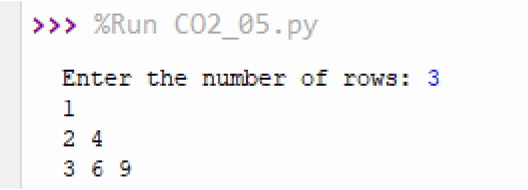
PROGRAM NO: 5

DATE:1/12/2021

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

```
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



```
>>> %Run CO2_05.py

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

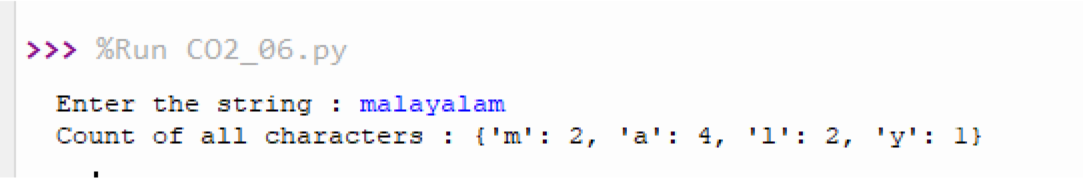
PROGRAM NO: 6

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



```
>>> %Run CO2_06.py

  Enter the string : malayalam
  Count of all characters : {'m': 2, 'a': 4, 'l': 2, 'y': 1}
```

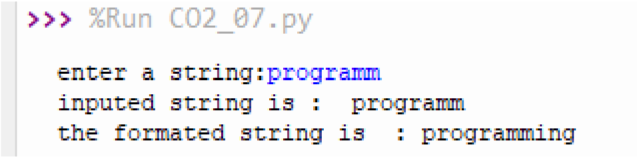
PROGRAM NO: 7

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

OUTPUT



```
>>> %Run CO2_07.py

  enter a string:programm
  inputed string is :  programm
  the formated string is  : programming
```

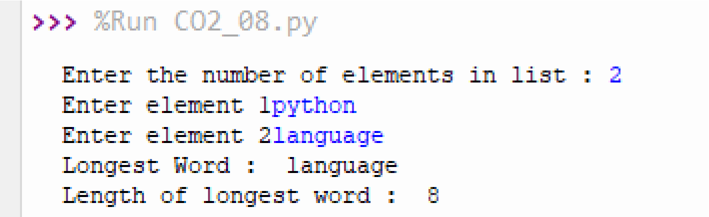
PROGRAM NO: 8

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



```
>>> %Run CO2_08.py

  Enter the number of elements in list : 2
  Enter element 1python
  Enter element 2language
  Longest Word :  language
  Length of longest word :  8
```

PROGRAM NO: 9

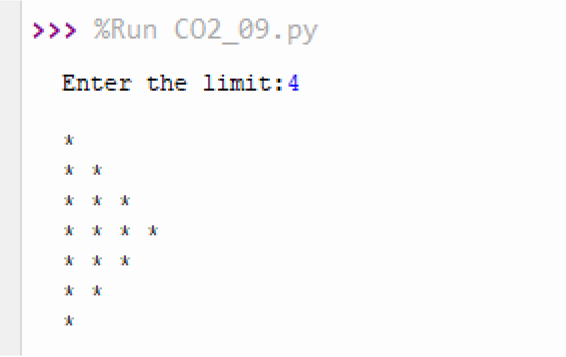
DATE:8/12/2021

AIM: Construct following pattern using nested loop

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

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

OUTPUT



```
>>> %Run CO2_09.py

  Enter the limit:4

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

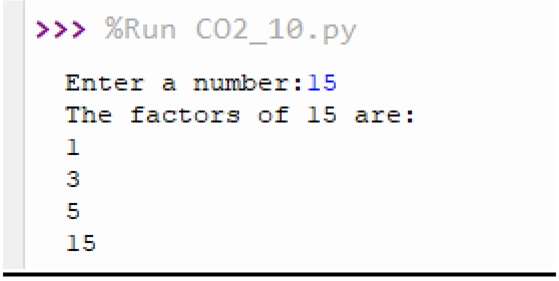
PROGRAM NO:10

DATE:8/12/2021

AIM: Generate all factors of a number.

```
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



```
>>> %Run CO2_10.py

  Enter a number:15
  The factors of 15 are:
  1
  3
  5
  15
```

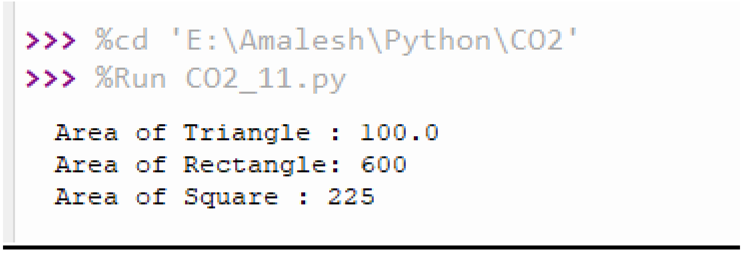
PROGRAM NO:11

DATE:8/12/2021

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

```
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



```
>>> %cd 'E:\Amalesh\Python\CO2'
>>> %Run CO2_11.py

  Area of Triangle : 100.0
  Area of Rectangle: 600
  Area of Square : 225
```

III. COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

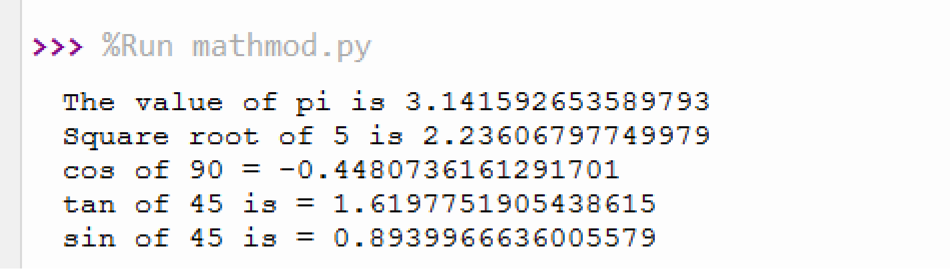
DATE:15/12/2021

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

A)MATH MODULE

```
# import math as m #importing math module as m
# print(m.pi) #prints the value of pi

import math as m
from math import pi,sqrt
print("The value of pi is",m.pi)
print("Square root of 5 is",sqrt(5))

print("cos of 90 =",m.cos(90))
print("tan of 45 is =",m.tan(45))
print("sin of 45 is =",m.sin(90))
```

OUTPUT

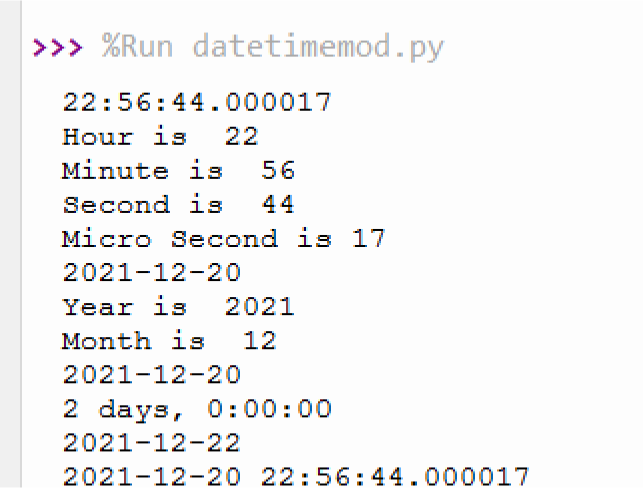


```
>>> %Run mathmod.py

  The value of pi is 3.141592653589793
  Square root of 5 is 2.23606797749979
  cos of 90 = -0.4480736161291701
  tan of 45 is = 1.6197751905438615
  sin of 45 is = 0.8939966636005579
```

B)DATE TIME MODULE

```
import datetime

t=datetime.time(22,56,44,17) #hour mint sec microsec
print(t)

print("Hour is ",t.hour)
print("Minute is ",t.minute)
print("Second is ",t.second)
print("Micro Second is",t.microsecond)

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

print("Year is ",d.year)
print("Month is ",d.month)

d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2) #timedelta class
print(td)
d2=d1+td #adding 2 days
print(d2)

dt=datetime.datetime.combine(d,t)
print(dt)
```

OUTPUT

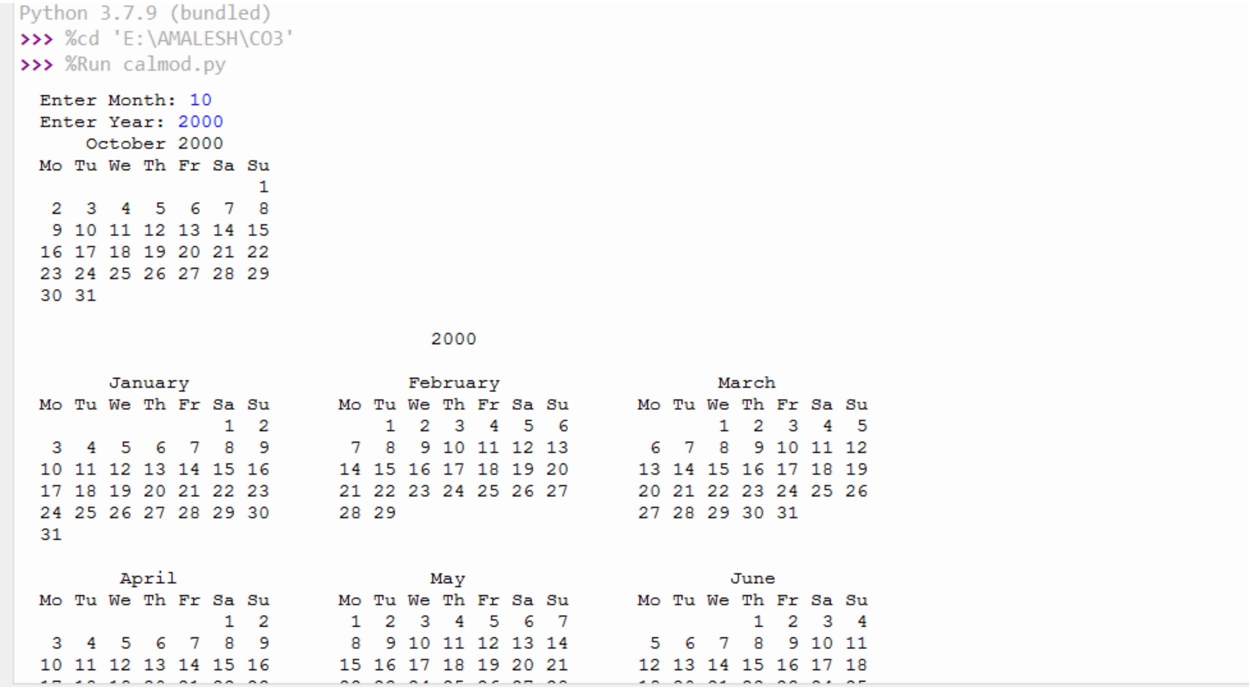


```
>>> %Run datetimemod.py
  22:56:44.000017
  Hour is  22
  Minute is  56
  Second is  44
  Micro Second is 17
  2021-12-20
  Year is  2021
  Month is  12
  2021-12-20
  2 days, 0:00:00
  2021-12-22
  2021-12-20 22:56:44.000017
```

C)CALENDAR MODULE

```
import calendar
mm=int(input("Enter Month: "))
yy=int(input("Enter Year: "))
print(calendar.month(yy,mm)) #printing the calendar of the specified month & year
print(calendar.calendar(yy)) #printing calendar of the specified year
```

OUTPUT

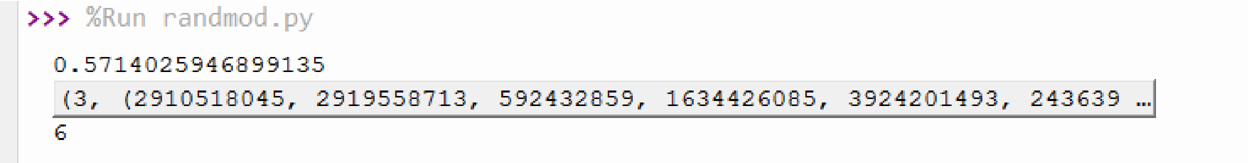


```
Python 3.7.9 (bundled)
>>> %cd 'E:\AMALESH\CO3'
>>> %Run calmod.py

 Enter Month: 10
 Enter Year: 2000
     October 2000
 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

                                   2000

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

D)RANDOM MODULE

```
i import random

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

print(random.getstate())

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

OUTPUT



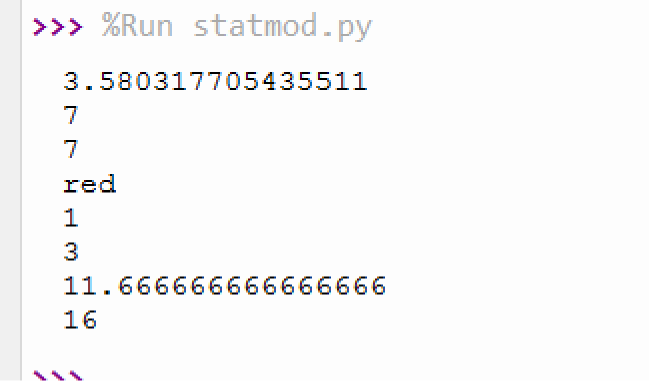
```
>>> %Run randmod.py

  0.5714025946899135
  (3, (2910518045, 2919558713, 592432859, 1634426085, 3924201493, 243639 …
  6
```

E)STATISTICS MODULE

```
# Import statistics Library
import statistics

a=[1, 3, 5, 7, 9, 11, 13]

# Calculate harmonic mean
print(statistics.harmonic_mean(a))
```

```
# Calculate average values
print(statistics.mean([1, 3, 5, 7, 9, 11, 13]))

# Calculate middle values
print(statistics.median([1, 3, 5, 7, 9, 11, 13]))

# Calculate the mode
print(statistics.mode(['red', 'green', 'blue', 'red']))
print(statistics.mode([1,2,3,1]))
print(statistics.mode([1, 3, 3, 3, 5, 7, 7, 9, 11]))
# Calculate the variance of an entire population
print(statistics.pvariance([1, 3, 5, 7, 9, 11]))
# Calculate the variance of an entire population
print(statistics.pvariance(a))
```

OUTPUT

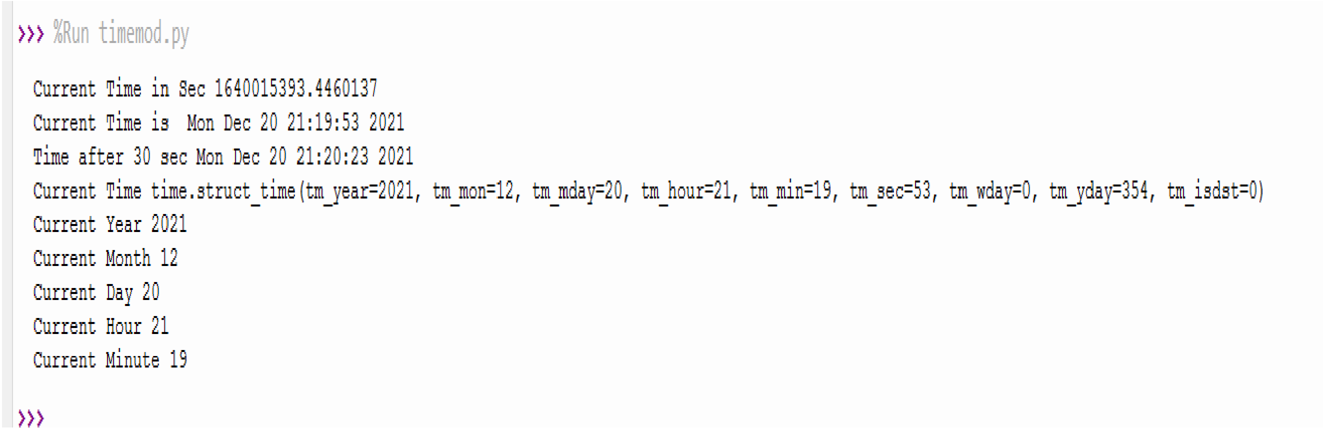


```
>>> %Run statmod.py

  3.580317705435511
  7
  7
  red
  1
  3
  11.666666666666666
  16
```

F)TIME MODULE

```
import time

print("Current Time in Sec",time.time())

print("Current Time is ",time.ctime()) #printing the current time

print("Time after 30 sec",time.ctime(time.time()+30)) #time after 30
sec

t=time.localtime()
print("Current 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 Minute",t.tm_min)
#print("Current Week",t.tm_week)
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:



```
>>> %Run timemod.py

 Current Time in Sec 1640015393.4460137
 Current Time is  Mon Dec 20 21:19:53 2021
 Time after 30 sec Mon Dec 20 21:20:23 2021
 Current Time time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=21, tm_min=19, tm_sec=53, tm_wday=0, tm_yday=354, tm_isdst=0)
 Current Year 2021
 Current Month 12
 Current Day 20
 Current Hour 21
 Current Minute 19

>>>
```

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

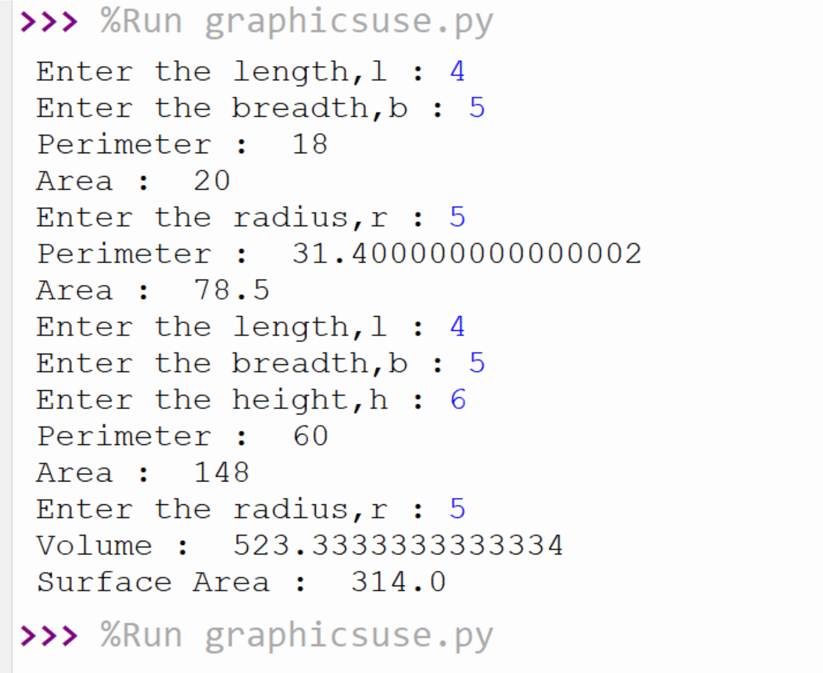
cuboid.py

```
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

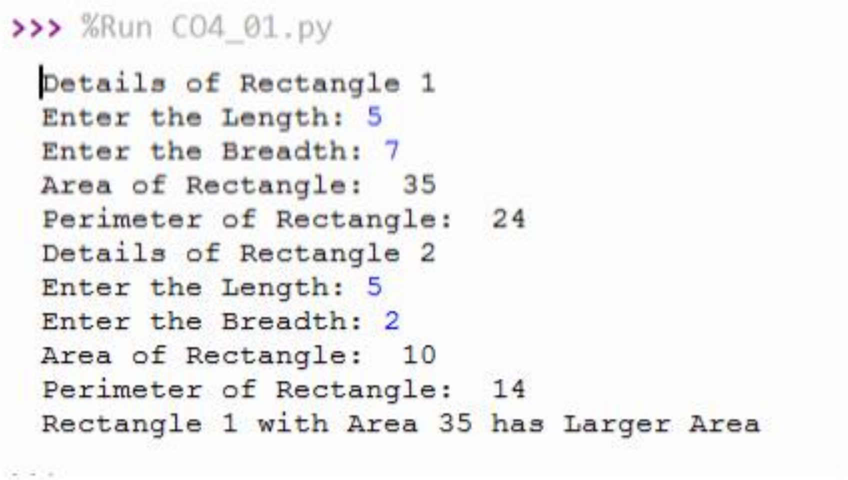
DATE:9/1/2022

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):
        self.length=int(input("Enter the Length: "))
        self.breadth=int(input("Enter the Breadth: "))
        self.area=self.length*self.breadth
        self.perimeter=2*(self.length+self.breadth)
    def display(self):
        print("Area of Rectangle: ",self.area)
        print("Perimeter of Rectangle: ",self.perimeter)


print("Details of Rectangle 1")
p1=Rectangle()
p1.display()
print("Details of Rectangle 2")
p2=Rectangle()
p2.display()
```

```
if p1.area>p2.area:

    print("Rectangle 1 with Area", p1.area, "has Larger Area")

else:

    print("Rectangle 2 with Area",p2.area,"has Larger Area")
```

OUTPUT



```
>>> %Run CO4_01.py

 Details of Rectangle 1
 Enter the Length: 5
 Enter the Breadth: 7
 Area of Rectangle:  35
 Perimeter of Rectangle:  24
 Details of Rectangle 2
 Enter the Length: 5
 Enter the Breadth: 2
 Area of Rectangle:  10
 Perimeter of Rectangle:  14
 Rectangle 1 with Area 35 has Larger Area
```

PROGRAM NO: 2

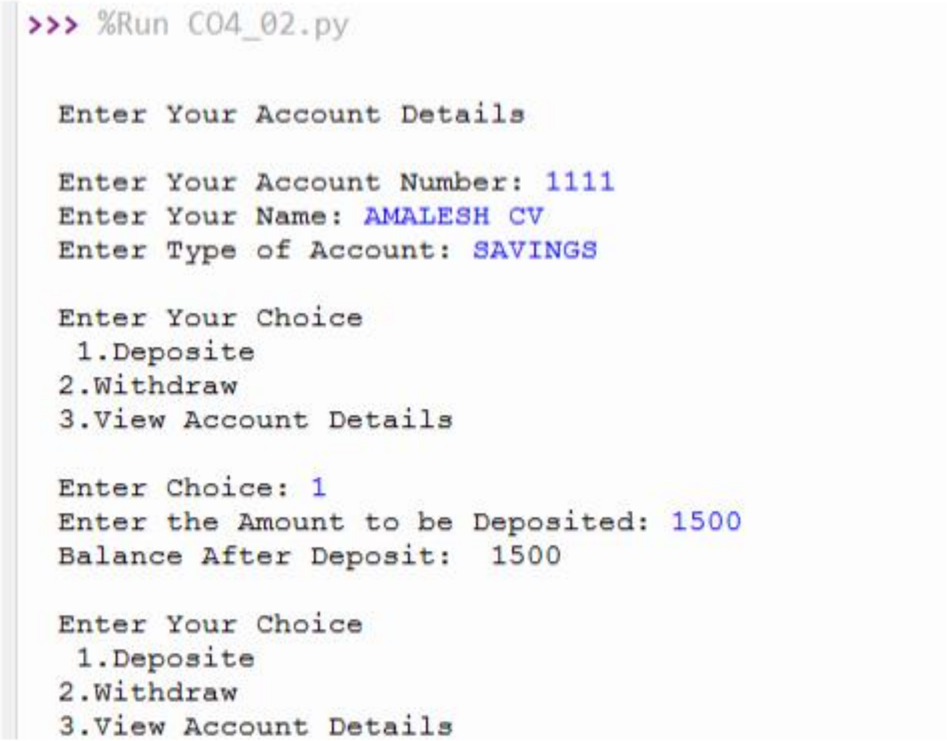
DATE:9/1/2022

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.

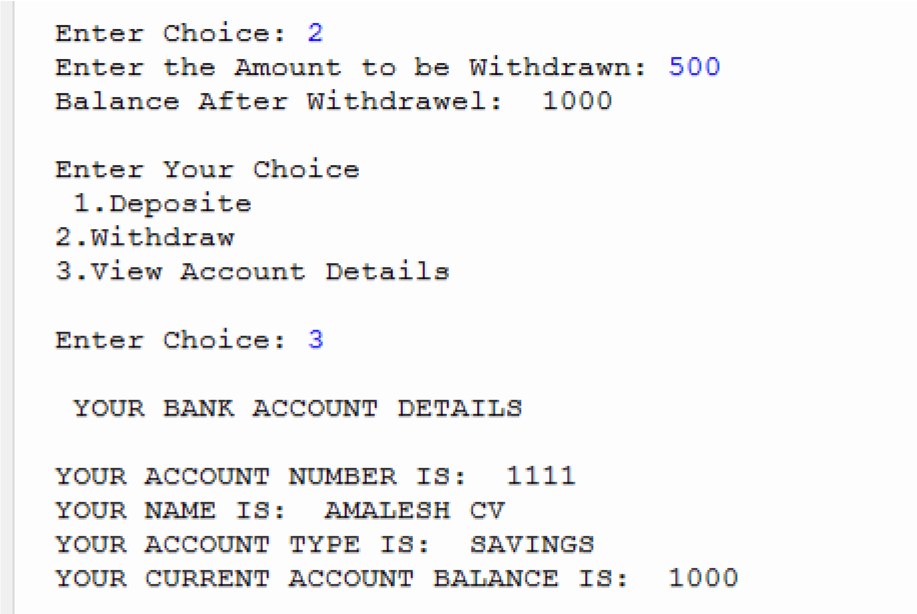
```
class Bank:
    def __init__(self):
        self.acbal=0

    def details(self):
        print("\nEnter Your Account Details\n")
        self.acno=int(input("Enter Your Account Number: "))
        self.name=input("Enter Your Name: ")
        self.actype=input("Enter Type of Account: ")

    def display(self):
        print("\n YOUR BANK ACCOUNT DETAILS \n")
        print("YOUR ACCOUNT NUMBER IS: ",self.acno)
        print("YOUR NAME IS: ",self.name)
        print("YOUR ACCOUNT TYPE IS: ",self.actype)
        print("YOUR CURRENT ACCOUNT BALANCE IS: ",self.acbal)

    def deposit(self):
        self.amount=int(input("Enter the Amount to be Deposited: "))
        self.acbal=self.acbal+self.amount
        print("Balance After Deposit: ",self.acbal)

    def withdraw(self):
        self.amount=int(input("Enter the Amount to be Withdrawn: "))
        self.acbal=self.acbal-self.amount
        print("Balance After Withdrawel: ",self.acbal)
```

```
B=Bank()
B.details()
x=1
while(x!=0):
        print("\nEnter Your Choice \n 1.Deposite\n2.Withdraw\n3.View Account
Details\n")
        x=int(input("Enter Choice: "))

        if x==1:
            B.deposit()
        elif x==2:
            B.withdraw()
        elif x==3:
            B.display()
        else:
            print("\n Invalid Operation")
```

OUTPUT



```
>>> %Run CO4_02.py

  Enter Your Account Details

  Enter Your Account Number: 1111
  Enter Your Name: AMALESH CV
  Enter Type of Account: SAVINGS

  Enter Your Choice
   1.Deposite
  2.Withdraw
  3.View Account Details

  Enter Choice: 1
  Enter the Amount to be Deposited: 1500
  Balance After Deposit:  1500

  Enter Your Choice
   1.Deposite
  2.Withdraw
  3.View Account Details
```



```
Enter Choice: 2
Enter the Amount to be Withdrawn: 500
Balance After Withdrawel:  1000

Enter Your Choice
 1.Deposite
2.Withdraw
3.View Account Details

Enter Choice: 3

 YOUR BANK ACCOUNT DETAILS

YOUR ACCOUNT NUMBER IS:  1111
YOUR NAME IS:  AMALESH CV
YOUR ACCOUNT TYPE IS:  SAVINGS
YOUR CURRENT ACCOUNT BALANCE IS:  1000
```

PROGRAM NO: 3

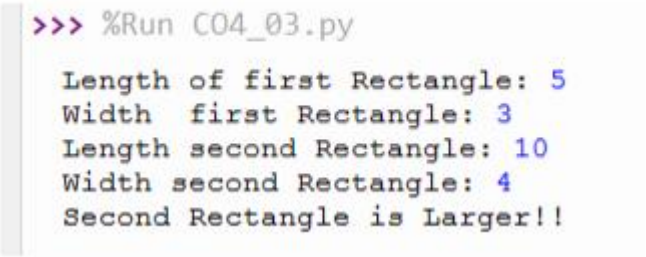
DATE:9/1/2022

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

```
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 first Rectangle: "))
b1=int(input("Width   first Rectangle: "))
r1=rectangle(a1,b1)

a2=int(input("Length second Rectangle: "))
```

```
b2=int(input("Width second Rectangle: "))

r2=rectangle(a2,b2)

if(r1<r2):

        print("Second Rectangle is Larger!!")

else:

        print("First Rectangle is Larger!!")
```

OUTPUT



```
>>> %Run CO4_03.py

  Length of first Rectangle: 5
  Width  first Rectangle: 3
  Length second Rectangle: 10
  Width second Rectangle: 4
  Second Rectangle is Larger!!
```

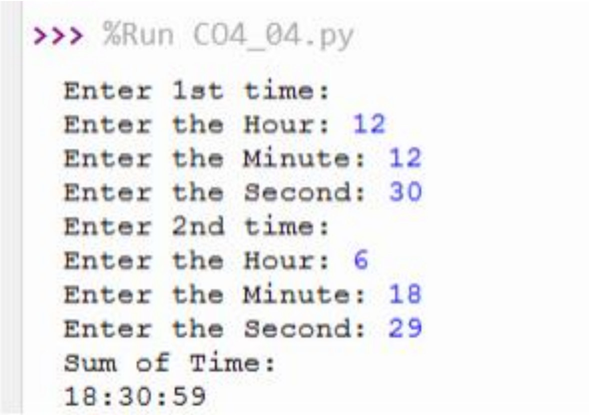
PROGRAM NO: 4

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
        if(hour>24):
            hour=hour-24
        return hour,minute,second

print("Enter 1st 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 2nd 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("Sum of Time: ")
print(hr,end=":")
print(min,end=":")
print(sec,end=" ")
```

OUTPUT



```
>>> %Run CO4_04.py

  Enter 1st time:
  Enter the Hour: 12
  Enter the Minute: 12
  Enter the Second: 30
  Enter 2nd time:
  Enter the Hour: 6
  Enter the Minute: 18
  Enter the Second: 29
  Sum of Time:
  18:30:59
```

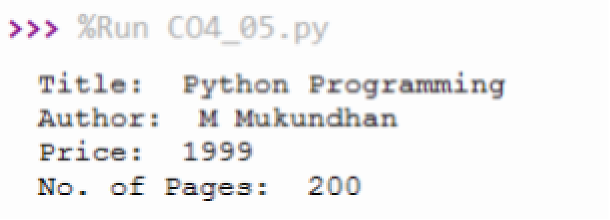
PROGRAM NO: 5

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 __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","M Mukundhan",1999,200)
p.display()
```

OUTPUT



```
>>> %Run CO4_05.py

  Title:  Python Programming
  Author:  M Mukundhan
  Price:  1999
  No. of Pages:  200
```

V. COURSE OUTCOME 5(CO5)

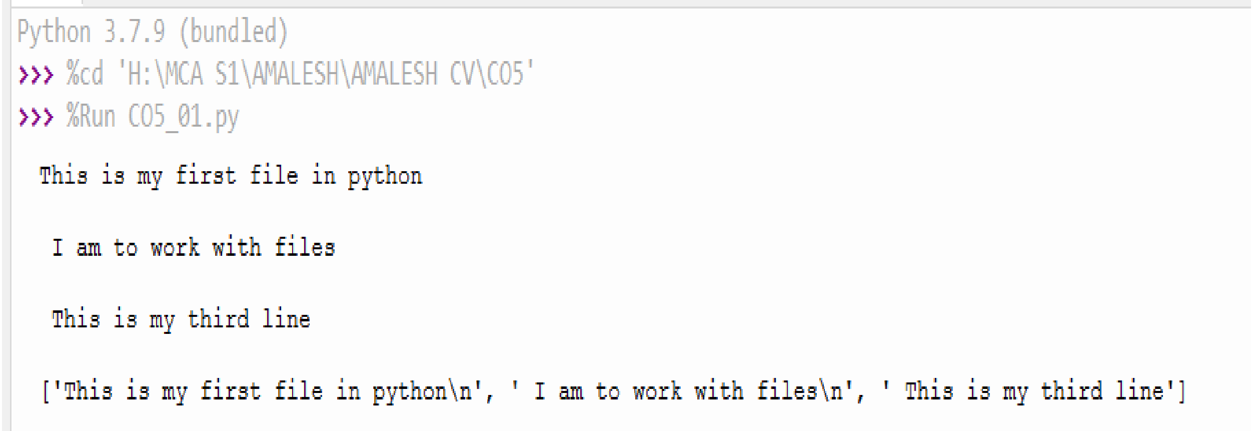
PROGRAM NO: 1

DATE:30/1/2022

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

```
f1=open("firstfile.txt","w")     #write mode
f1.write("This is my first file in python\n I am to work with files\n This is my third line")   #writing to the file
f1.close()

f1=open("firstfile.txt","r")
f1.seek(0,0)
ff=f1.readlines()
for x in range (0,len(ff)):
      print(ff[x])

print()
print(ff)
f1.close
```

OUTPUT



```
Python 3.7.9 (bundled)
>>> %cd 'H:\MCA S1\AMALESH\AMALESH CV\CO5'
>>> %Run CO5_01.py

 This is my first file in python

  I am to work with files

  This is my third line

 ['This is my first file in python\n', ' I am to work with files\n', ' This is my third line']
```

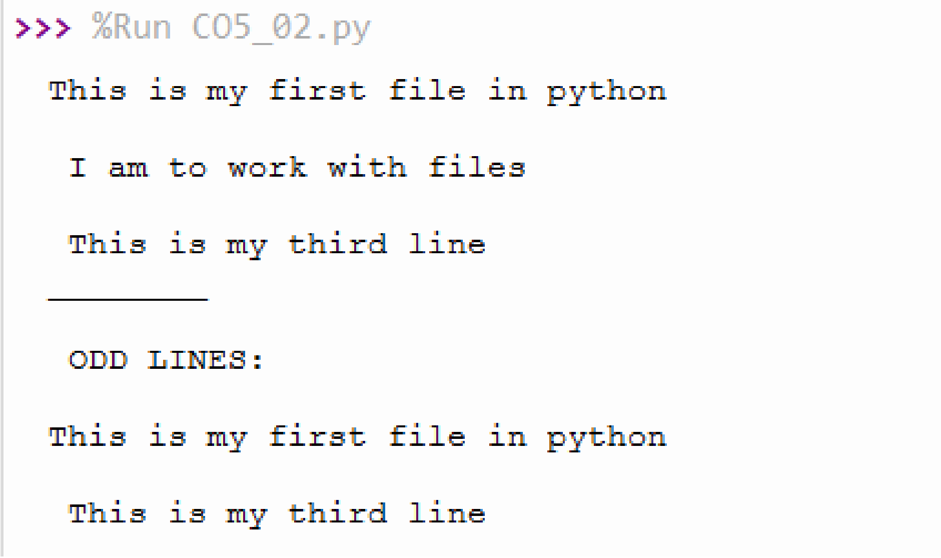
PROGRAM NO: 2

DATE:30/1/2022

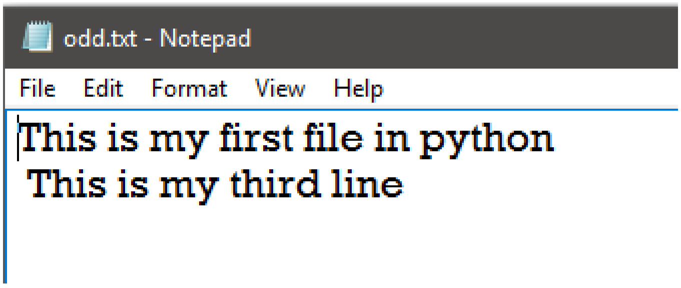
AIM: Python program to copy odd lines of one file to other

```
f1=open("firstfile.txt","r")

for x in f1:
    print(x)
print("________")
f1.seek(0,0)
ff=f1.readlines()

f2=open("odd.txt","w") #creating a new file for writing the odd lines of file 1
print("\n ODD LINES: \n")
for x in range(0,len(ff)):
    if(x%2==0):
        print(ff[x])
        f2.write(ff[x])

f2.close()
```

OUTPUT



```
>>> %Run CO5_02.py
  This is my first file in python

   I am to work with files

   This is my third line
  ________

   ODD LINES:

  This is my first file in python

   This is my third line
```



```
This is my first file in python
 This is my third line
```

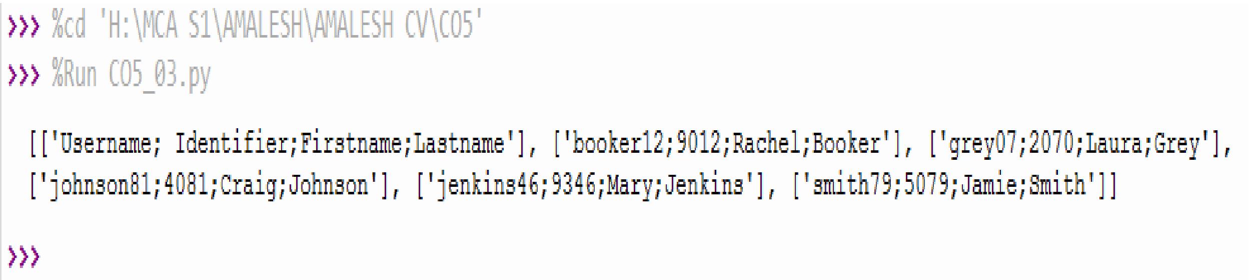
PROGRAM NO: 3

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



```
>>> %cd 'H:\MCA S1\AMALESH\AMALESH CV\CO5'
>>> %Run CO5_03.py

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

>>>
```

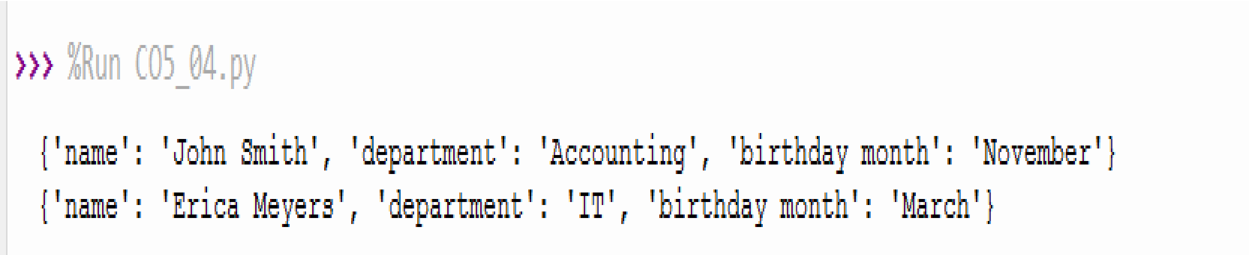
PROGRAM NO: 4

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 CO5_04.py

  {'name': 'John Smith', 'department': 'Accounting', 'birthday month': 'November'}
  {'name': 'Erica Meyers', 'department': 'IT', 'birthday month': 'March'}
```

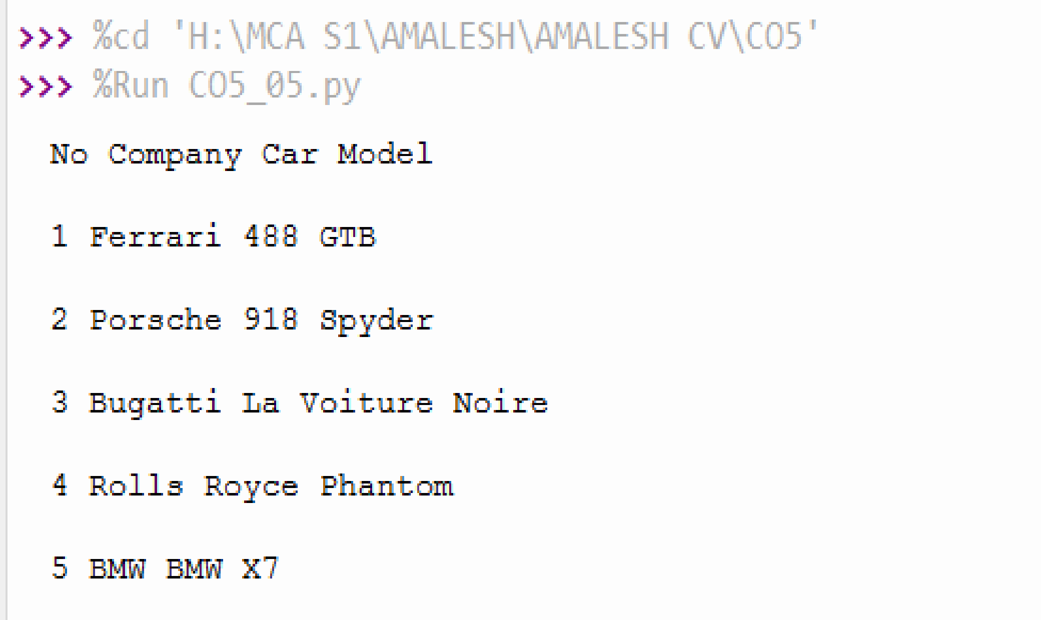
PROGRAM NO: 5

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



```
>>> %cd 'H:\MCA S1\AMALESH\AMALESH CV\CO5'
>>> %Run CO5_05.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
```