COURSE OUTCOME 1 (CO1)

PROGRAM NO: 1 **DATE : 03/11/2021**

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

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Build, execution, and debugging tools
- Some form of source control

Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly. General Editors and IDEs with Python Support

1.Eclipse + PyDev

PyDev, which enables Python debugging, code completion, and an interactive Python console. Installing PyDev into Eclipse is easy: from Eclipse, select Help, Eclipse Marketplace, then search for PyDev.

2.Sublime Text

Written by a Google engineer with a dream for a better text editor, Sublime Text is an extremely popular code editor. Supported on all platforms, it has built-in support for Python code editing and a rich set of extensions that extend the syntax and editing features.

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.

PROGRAM NO: 2 **DATE : 08/11/2021**

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

PROGRAM

```
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) or (i%400==0) and (i%100==0):
                print(i,end=" ")
```

OUTPUT

```
>>> %Run CO1_2.py
 enter start year:1990
 enter end year:2023
 Leap years are: 1992 1996 2000 2004 2008 2012 2016 2020
>>>
```

PROGRAM NO: 3 **DATE : 10/11/2021**

AIM : 3.1. Generate positive list of numbers from a given list of integers

PROGRAM

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

```
>>> %Run CO1_3.1.py
 [23, 55, 79, 34]
>>>
```

AIM : 3.2. Write a program to find the Square of N number

PROGRAM

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

OUTPUT

```
>>> %Run CO1_3.2.py
 enter limit:10
 Square of N numbers :  [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
>>>
```

AIM : 3.3. Form a list of vowels selected from a given word

PROGRAM

```
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

```
>>> %Run CO1_3.3.py
 Enter the word :hi hello welcome
 The original string is : hi hello welcome
 The vowels are : ['i'] ['e'] ['o'] ['e'] ['o'] ['e']
>>>
```

AIM : 3.4. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

PROGRAM

```
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

```
>>> %Run CO1_3.4.py
 Enter a word:welcome
 Ordinal values corresponding to each element is:
 w:119 e:101 l:108 c:99 o:111 m:109 e:101
>>>
```

PROGRAM NO: 4 **DATE : 15/11/2021**

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

PROGRAM

```
str1=input("Enter any sentence: ")
wordlist=str1.split()
count= []
for i in wordlist:
        count.append(wordlist.count(i))
print("count of the occurrence:" + str(list(zip(wordlist, count))))
```

OUTPUT

```
>>> %Run CO1_4.py
 Enter a String :this is a python programming language
 count of the occurrence:[('this', 1), ('is', 1), ('a', 1), ('python', 1), ('programming', 1
 ), ('language', 1)]
>>>
```

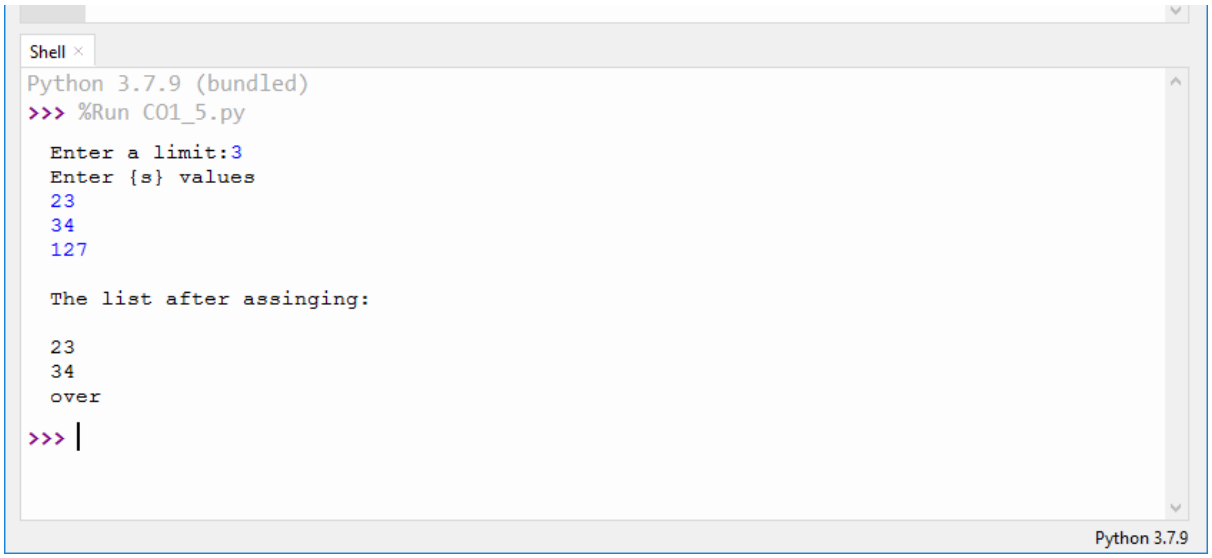
PROGRAM NO: 5 **DATE : 15/11/2021**

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

PROGRAM

```
n=[]
s=int(input("Enter a limit:  "))
print("Enter  list values")
for i in range(0,s):
        n.append(int(input()))
print("The list after assinging: ",end=" ")
for i in range(0,len(n)):
        if n[i]<=100:
                print(n[i] ,end=" ")
        else:
                print("over")
```

OUTPUT



PROGRAM NO: 6 **DATE : 17/11/2021**

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

PROGRAM

```
list1 = ["a", "b", "c","c","b"", "a", "c","c","b"]]

occ = list1.count("a")

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

OUTPUT

```
>>> %Run CO1_6.py
  count of occurences of a: 2
>>>
```

PROGRAM NO: 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.

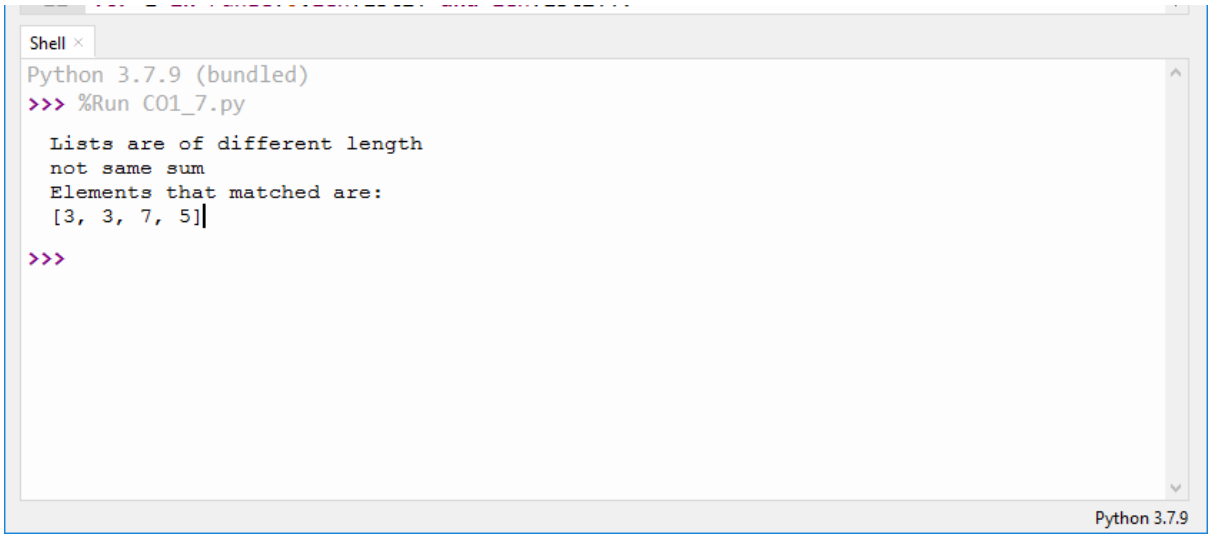
(c) whether any value occur in both.

PROGRAM

```
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(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



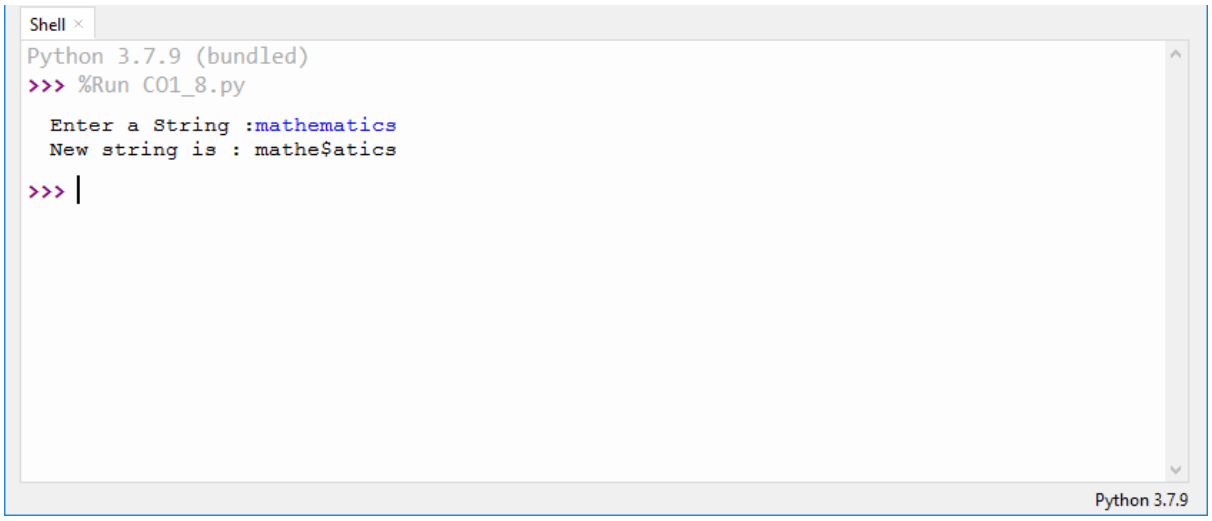
PROGRAM NO: 8 **DATE : 22/11/2021**

AIM : Get a string from an input string where all occurrences of first character replaced with '$', except first character.

PROGRAM

```
str1 = input("Enter a String :")
ch = str1[0]
str1 = str1.replace(ch, '$')
str1 = ch+ str1[1:]
print("New string is :",str1)
```

OUTPUT



PROGRAM NO: 9 **DATE : 22/11/2021**

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

PROGRAM

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

OUTPUT

```
>>> %Run CO1_9.py
  Enter a string:python
  New string: nythop
>>>
```

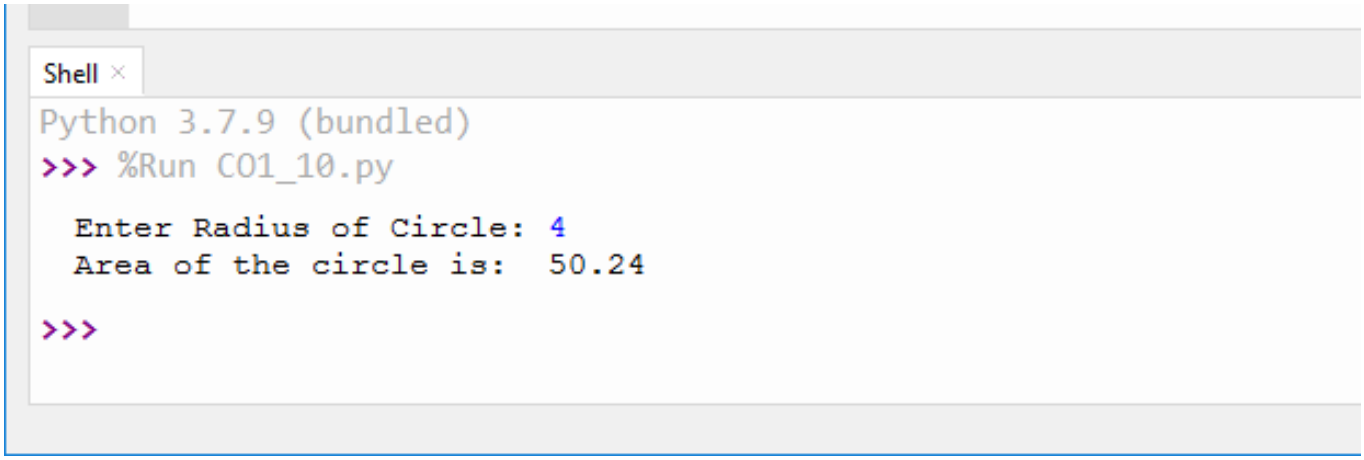
PROGRAM NO: 10 **DATE : 24/11/2021**

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

PROGRAM

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

OUTPUT



PROGRAM NO: 11 **DATE : 24/11/2021**

AIM : Write a program to find biggest of 3 numbers entered.

PROGRAM

```
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

```
>>> %Run CO1_11.py
 Enter 1st number : 5
 Enter 2nd number : 8
 Enter 3rd number : 10
 The largest number is 10
>>>
```

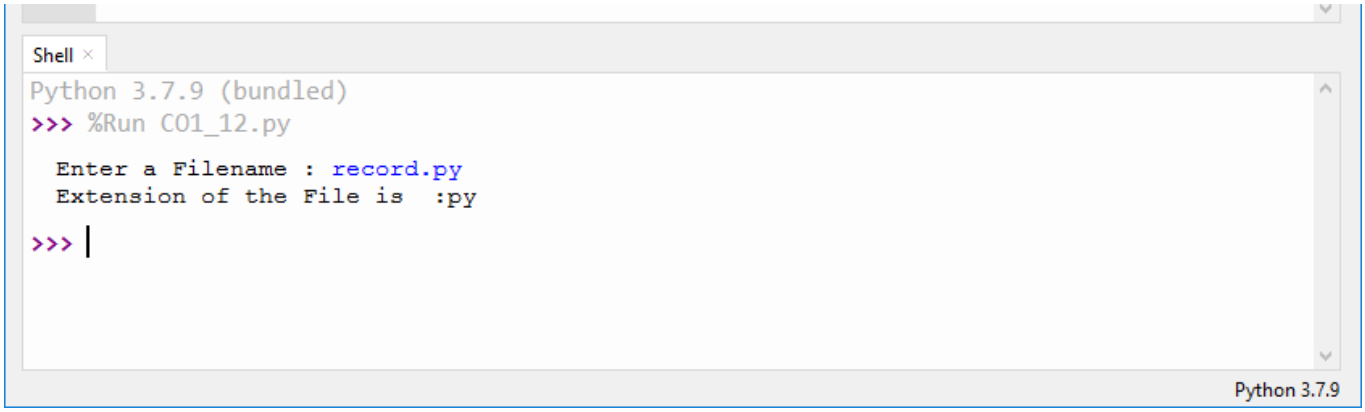
PROGRAM NO: 12 **DATE : 24/11/2021**

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

PROGRAM

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

OUTPUT



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

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

OUTPUT

```
>>> %Run CO1_13.py
  Enter the color : blue
  Enter the color : red
  Enter the color : green
  ['blue', 'red', 'green']
  First color : blue
  Second color : green
>>>
```

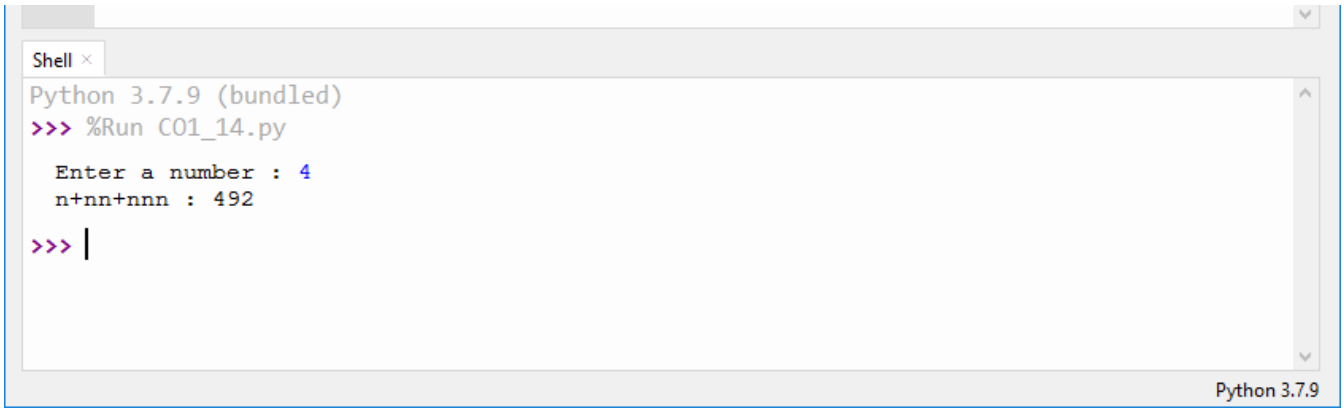
PROGRAM NO: 14 **DATE : 29/11/2021**

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

PROGRAM

```
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



PROGRAM NO: 15 **DATE : 29/11/2021**

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

PROGRAM

```
list1={"White","pink","Red","Blue"}
list2={"Red","Green","pink"}
print(list1.difference(list2))
```

OUTPUT

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

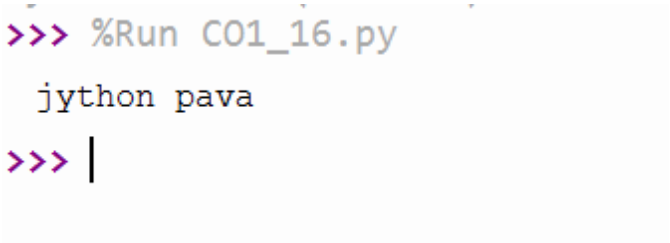
PROGRAM NO: 16 **DATE : 29/11/2021**

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

PROGRAM

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

OUTPUT



PROGRAM NO: 17 **DATE : 01/12/2021**

AIM : Sort dictionary in ascending and descending order.

PROGRAM

```
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

```
>>> %Run CO1_17.py
 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}
>>>
```

PROGRAM NO: 18 **DATE : 01/12/2021**

AIM : Write a program to merge two dictionaries.

PROGRAM

```
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}
>>>
```

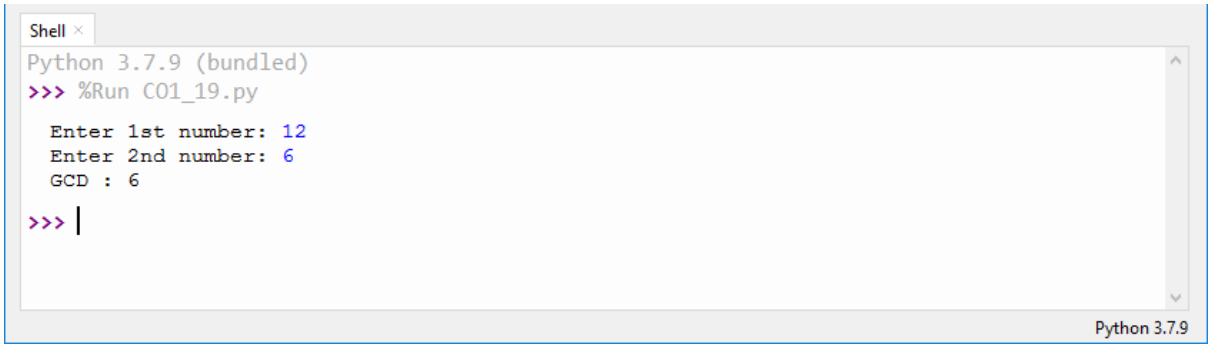
PROGRAM NO: 19 **DATE : 01/12/2021**

AIM : Write a program to find GCD of 2 numbers.

PROGRAM

```
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



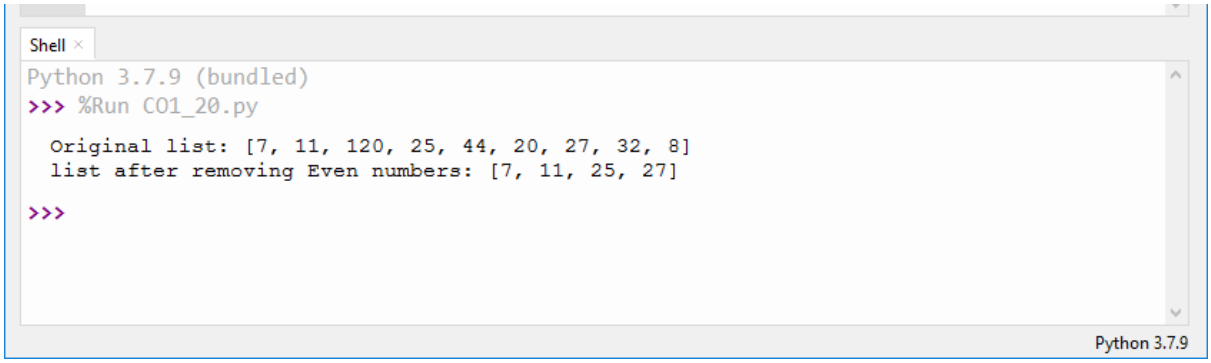
PROGRAM NO: 20 **DATE : 01/12/2021**

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

PROGRAM

```
num = [7,11, 120, 25, 44, 20, 27,32,8]
print( "Original list:",num)
num = [x for x in num if x%2!=0]
print("List after removing even numbers:",num)
```

OUTPUT



COURSE OUTCOME 2 (CO2)

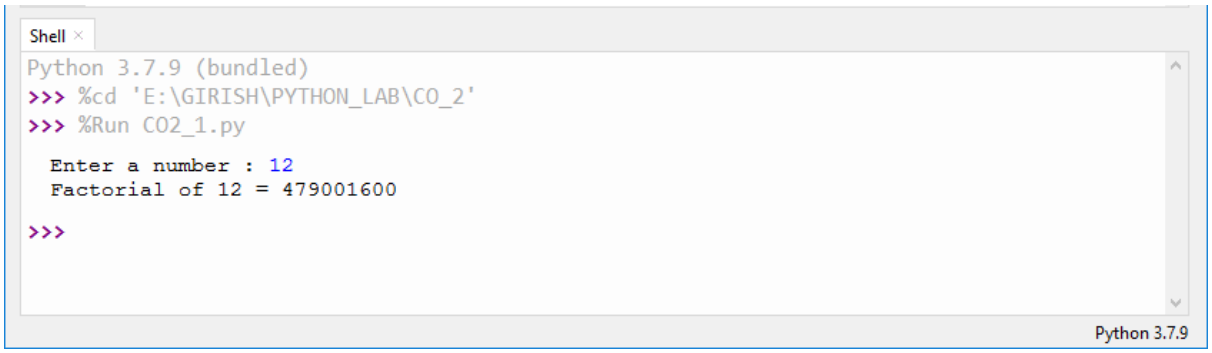
PROGRAM NO: 1 **DATE : 06/12/2021**

AIM : Write a program to find the factorial of a number.

PROGRAM

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

OUTPUT



PROGRAM NO: 2 **DATE : 06/12/2021**

AIM : Write a program to generate Fibonacci series of N terms.

PROGRAM

```
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_2.py
 Enter the limit : 5
 Fibonacci Series : 0 1 1 2 3
>>>
```

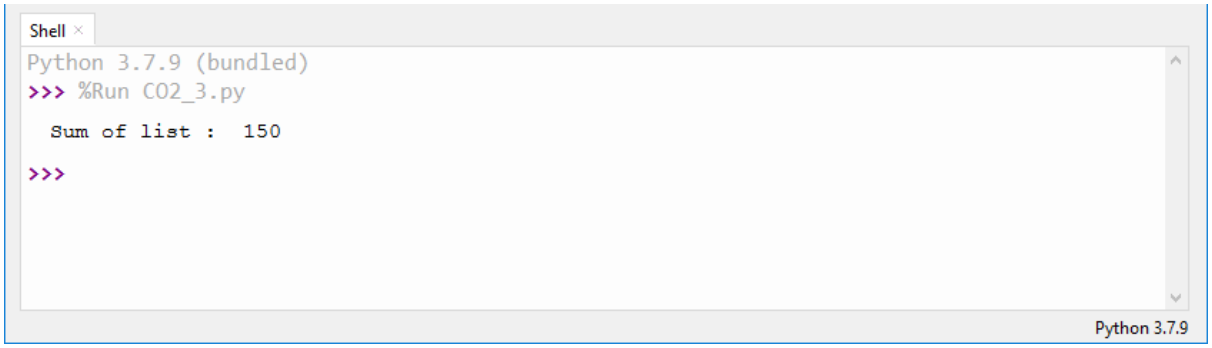
PROGRAM NO: 3 **DATE : 06/12/2021**

AIM : Write a program to find the sum of all items in a list.

PROGRAM

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

total=sum(list1)

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

OUTPUT



PROGRAM NO: 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.

PROGRAM

```
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_4.py
 1024 1156 1296 1444 1600 1764 1936 2116 2
 304 2500 2704 2916 3136 3364 3600 3844 40
 96 4356 4624 4900 5184 5476 5776 6084 640
 0 6724 7056 7396 7744 8100 8464 8836 9216
 9604
>>>
```

PROGRAM NO: 5 **DATE : 06/12/2021**

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

PROGRAM

```
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

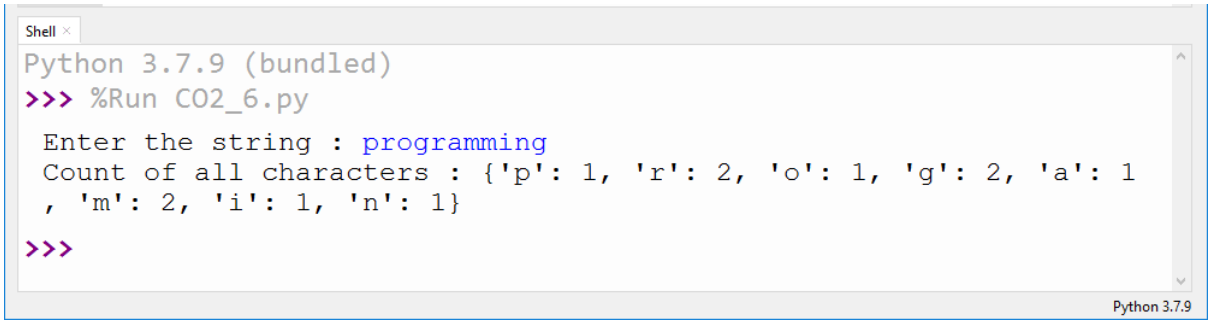
PROGRAM NO: 6 **DATE : 06/12/2021**

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

PROGRAM

```
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



PROGRAM NO: 7 **DATE : 08/12/2021**

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

PROGRAM

```
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_7.py
  enter a string:hang
  inputed string is: hang
  the formated string is: hanging
>>> %Run CO2_7.py
  enter a string:ly
  inputed string is: ly
  the formated string is: lying
>>>
```

```
>>> %Run CO2_7.py
  enter a string:Cry
  inputed string is: Cry
  the formated string is: Crying
>>>
```

PROGRAM NO: 8 **DATE : 08/12/2021**

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

PROGRAM

```
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_8.py
  Enter the number of elements in list:2
  Enter element 1python
  Enter element 2programming
  Longest Word: programming
  Length of longest word : 11
>>>
```

PROGRAM NO: 9 **DATE : 08/12/2021**

AIM : Construct following pattern using nested loop

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

PROGRAM

```
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_9.py
 Enter the limit:4
 *
 * *
 * * *
 * * * *
 * * *
 * *
 *
>>>
```

PROGRAM NO: 10 **DATE : 08/12/2021**

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

PROGRAM

```
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 :10
  The factors of 10 are:
  1
  2
  5
  10
>>>
```

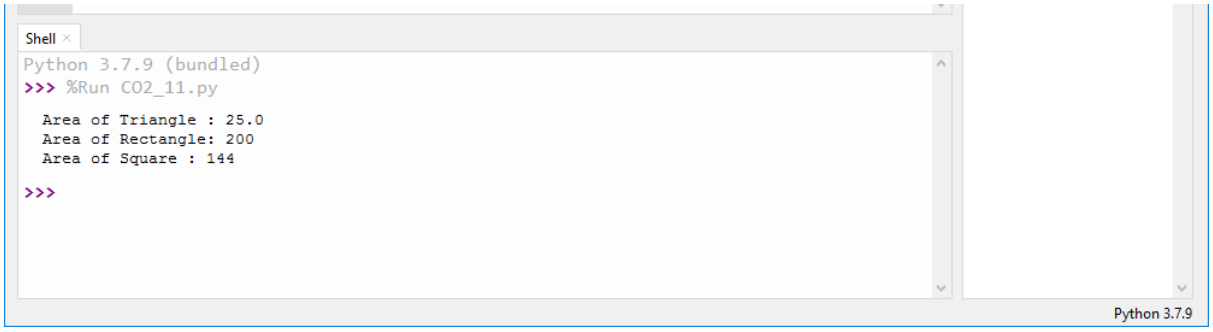
PROGRAM NO: 11 **DATE : 08/12/2021**

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

PROGRAM

```
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



COURSE OUTCOME 3 (CO3)

PROGRAM NO: 1 **DATE : 13/12/2021**

AIM : Work with built-in packages.

A) **Module math**
B) **Module time**
C) **Module calendar**
D) **Module random**
E) **Module statistics**

A) Module math

PROGRAM

```
import math
print(math.pi)
print(".....\n")

import math as m
print(m.pi)
print(".....\n")

from math import pi,sqrt
print("Value of pi is ",pi)
print("Value of square root is ",sqrt(9))
print(".....\n")

from math import sin,cos,tan
print("Value of sin(90) is ",sin(90))
print("Value of cos(90) is ",cos(90))
print(math.cos(90))
print("Value of tan(90) is ",tan(90))
print(".....\n")
```

OUTPUT

```
>>> %Run 'module math.py'
 3.141592653589793
 ....................

 3.141592653589793
 ....................

 Value of pi is  3.141592653589793
 Value of square root is  3.0
 ....................

 Value of sin(90) is  0.8939966636005579
 Value of cos(90) is  -0.4480736161291701
 -0.4480736161291701
 Value of tan(90) is  -1.995200412208242
 ....................

>>>
```

B) Module time

PROGRAM

```
import time
print("Current time in second : ",time.time())
print("Current time : ",time.ctime())
print("Current time after 30 seconds : ",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 week day:",t.tm_wday)
print("current Hour:",t.tm_hour)
print("current Minute:",t.tm_min)
print("current Second:",t.tm_sec)
```

OUTPUT

```
>>> %Run 'module time.py'
 Current time in second :  1640014835.8148754
 Current time :  Mon Dec 20 21:10:35 2021
 Current time after 30 seconds :  Mon Dec 20 21:11:05 2021
 time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=21, tm_min=10,
 tm_sec=35, tm_wday=0, tm_yday=354, tm_isdst=0)
 current year: 2021
 current month: 12
 current day: 20
 current week day: 0
 current Hour: 21
 current Minute: 10
 current Second: 35
>>>
```

C) Module calendar

PROGRAM

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

OUTPUT

```
>>> %Run 'module calendar.py'

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

                                  2015

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

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

>>>
```

D) Module random

PROGRAM

```
import random
mylist = ["apple", "banana", "cherry"]
print(random.choice(mylist))        #Returns a random element from the given sequence
print(random.choices(mylist, k=2))
print(random.sample(mylist, k=2))   #Return a list that contains any 2 of the items from a list:
random.shuffle(mylist)
print(mylist)                       #Takes a sequence and returns the sequence in a random order
print(random.randrange(3, 9))       #Return a number between 3 and 9
```

OUTPUT

```
>>> %Run 'module random.py'
 apple
 ['apple', 'cherry']
 ['apple', 'banana']
 ['cherry', 'banana', 'apple']
 4

>>>
```

E) Module statistics

PROGRAM

```
import statistics
print(statistics.mean([10,20,30,40,50,60]))    #To calculate the mean of given numbers
print(statistics.median([10,20,30]))    #To find the median of given numbers
print(statistics.harmonic_mean([10,20,30,40,50,60]))    #To calculate the harmonic mean of given
numbers
```

OUTPUT

```
>>> %Run 'module statistics (1).py'
 35
 20
 24.489795918367346
>>>
```

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)

PROGRAM

Package graphics

(1) __init__.py

(2) circle.py

```
def perimeter(r):
   print ("Perimeter of the circle : ",2*3.14*r)

def area(r):
   print ("Area of the circle : ",3.14*r*r)
```

(3) rectangle.py

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

def area(l,b):
    print ("Area of the rectangle : ",l*b)
```

(4) Subpackage ThreeDgraphics

i. __init__.py

ii. cuboid.py

```
def perimeter(l,b,h):
   print ("Perimeter of the cuboid : ",4*(l+b+h))
```

```
def area(l,b,h):
    print ("Area of the cuboid : ",2*l*b+2*l*h+2*h*b)
```

iii. sphere.py

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

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 of rectangle,l : "))
b=int(input("Enter the breadth of rectangle,b : "))
rectangle.perimeter(l,b)
rectangle.area(l,b)
print()

r=int(input("Enter the radius of circle,r : "))
circle.perimeter(r)
circle.area(r)
print()
```

```
l=int(input("Enter the length of cuboid,l : "))

b=int(input("Enter the breadth of cuboid,b : "))

h=int(input("Enter the height of cuboid,h : "))

cuboid.perimeter(l,b,h)

cuboid.area(l,b,h)

print()


r=int(input("Enter the radius of sphere,r : "))

sphere.volume(r)

sphere.area(r)
```

OUTPUT

```
>>> %Run graphicsuse.py
 Enter the length of rectangle,l : 4
 Enter the breadth of rectangle,b : 5
 Perimeter of the rectangle :  18
 Area of the rectangle :  20

 Enter the radius of circle,r : 5
 Perimeter of the circle :  31.400000000000002
 Area of the circle :  78.5

 Enter the length of cuboid,l : 4
 Enter the breadth of cuboid,b : 5
 Enter the height of cuboid,h : 6
 Perimeter of the cuboid :  60
 Area of the cuboid :  148

 Enter the radius of sphere,r : 5
 Volume of the sphere :  523.3333333333334
 Surface Area of the sphere :  314.0
>>>
```

COURSE OUTCOME 4 (CO4)

PROGRAM NO: 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.

PROGRAM

```
class Rectangle:
    def __init__(self,length,breadth,ar):
        self.length=length
        self.breadth=breadth
        self.ar=0
    def area(self):
        self.ar=self.length*self.breadth
        #print("area=",self.ar)
        return (self.ar)
    def perimeter(self):
        self.perimeter=2*(self.length+self.breadth)
        #print(perimeter)
        return (self.perimeter)
    def display(self):
        print("Area=",self.ar)
        print("Perimeter=",self.perimeter)

R1=Rectangle(2,4,0)
R2=Rectangle(3,4,0)
R1.area()
R1.perimeter()
```

```
R2.area()

R2.perimeter()

print("Rectangle 1")

R1.display()

print("Rectangle 2")

R2.display()

if (R1.ar>R2.ar):

    print("Rectangle 1 is larger")

else:

    print("Rectangle 2 is larger")
```

OUTPUT

```
>>> %Run co4_1.py
 Rectangle 1
 Area= 8
 Perimeter= 12
 Rectangle 2
 Area= 12
 Perimeter= 14
 Rectangle 2 is larger
>>>
```

PROGRAM NO: 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.

PROGRAM

```
class bank:
    __acc_name=""
    __acc_no = ""
    __acc_type = ""
    __acc_balance = 0

    def __init__(self,a_name,a_no,a_type,a_balance):
        self.__acc_name = a_name
        self.__acc_no = a_no
        self.__acc_type = a_type
        self.__acc_balance = a_balance

    def deposite(self,a_deposit):
        print("Initial balance is  : ",self.__acc_balance)
        print("Deposite is  : ",a_deposit)
        self.__acc_balance += a_deposit
        print("Current balance is  : ",self.__acc_balance)

    def withdraw(self):
        print("Current balance is  : ",self.__acc_balance)
        self.amount = int(input("How much amount need to withdraw : "))
        if self.amount > self.__acc_balance:
            print("You don't have enough balance to withdraw !!")
            print("Current balance is  : ",self.__acc_balance)
        else:
            print(self.amount," is withrawed .")
            self.__acc_balance -= self.amount
            print("Current balance is  : ",self.__acc_balance)
```

```
    def acc_info(self):
        print("\n\n.....\n\n")
        print("Account holder name  :  ",self.__acc_name)
        print("Account number       :  ",self.__acc_no)
        print("Account type         :  ",self.__acc_type)
        print("Account Balance is   :  ",self.__acc_balance)
        print("\n\n.....\n\n")
def main():

    name  = input("Enter Account holder name : ")
    no    = input("Enter Account number       : ")
    atype = input("Enter Account type         : ")
    bal   = int(input("Enter Account initial balance : "))
    holder = bank(name,no,atype,bal)

    while(True):
        print("\n\n.....\n\n")
        opt = int(input("1)Deposite \n2)Withdraw \n3)Account info \n0)Exit\nChoose your option :: "))
        print("\n\n.....\n\n")
        if opt == 1:
            amount = int(input("Deposite amount : "))
            holder.deposite(amount)
        elif opt == 2:
            holder.withdraw()
        elif opt == 3:
            holder.acc_info()
        elif opt == 0:
            break
        else:
            print("Invalid Option !")
if __name__ == "__main__":
    while(True):
        main()
```

OUTPUT

```
Python 3.7.9 (bundled)
>>> %Run CO4_4.py
 Enter Account holder name : Girish
 Enter Account number        : 4425771
 Enter Account type           : savings
 Enter Account initial balance : 100


 .....................................................


 1)Deposite
 2)Withdraw
 3)Account info
 0)Exit
 Choose your option :: 1


 .....................................................
```

Python 3.7.9

Shell

```
 Deposite amount : 200
 Initial balance is  :  100
 Deposite is  :  200
 Current balance is  :  300


 .....................................................


 1)Deposite
 2)Withdraw
 3)Account info
 0)Exit
 Choose your option :: 2


 .....................................................


 Current balance is  :  300
 How much amount need to withdraw :
```

Python 3.7.9

Shell

```


 Current balance is  :  300
 How much amount need to withdraw : 50
 50  is withrawed .
 Current balance is  :  250


 .....................................................


 1)Deposite
 2)Withdraw
 3)Account info
 0)Exit
 Choose your option :: 3


 .....................................................
```

Shell

```
 .....................................................


 Account holder name  :   Girish
 Account number         :   4425771
 Account type             :   savings
 Account Balance is      :   250


 .....................................................



 .....................................................


 1)Deposite
 2)Withdraw
 3)Account info
 0)Exit
 Choose your option ::
```

PROGRAM NO: 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.

PROGRAM

```
class rectangle:
    def __init__(self,length,width):
        self.length=length
        self.width=width
    def __lt__(self,a2):
        area1=self.length*self.width
        area2=a2.length*a2.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

```
>>> %Run co4_3.py
 Enter the Details of Rectangle:1
 Length : 50
 Width : 60
 Enter the Details of Rectangle:2
 Length : 5
 Width : 6
 Rectangle 1 is larger!!
>>>
```

PROGRAM NO: 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

PROGRAM

```
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

```
Python 3.7.9 (bundled)
>>> %Run C04_4.py

  Enter 1 time:
  enter the hour:2
  enter the minute:23
   enter the second:34
  Enter  2 time:
  enter the hour:5
  enter the minute:20
  enter the second:30
  7:44:4

>>>
```

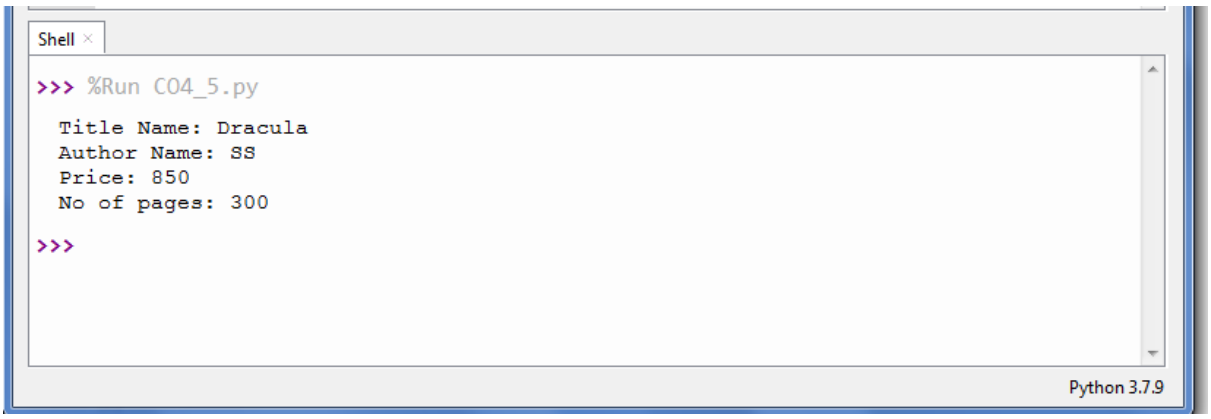
PROGRAM NO: 5 **DATE: 10/01/2022**

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.

PROGRAM

```
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(850,300,"K D")
s1.get("Dracula","SS")
s1.display()
s1.details()
```

OUTPUT


Shell
>>> %Run C04_5.py
Title Name: Dracula
Author Name: SS
Price: 850
No of pages: 300
>>>
Python 3.7.9

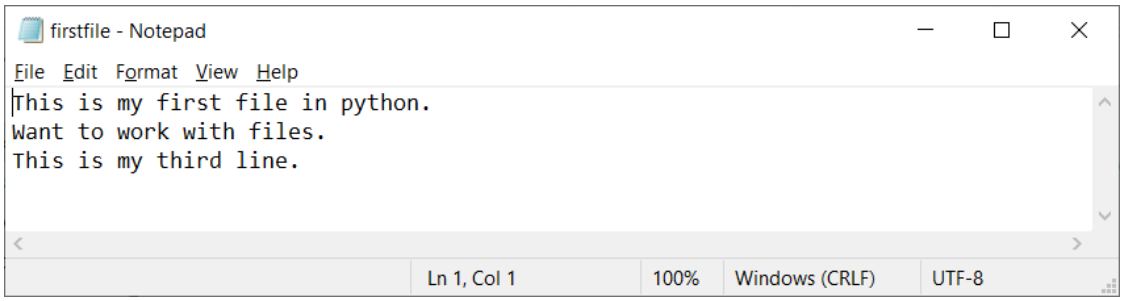
COURSE OUTCOME 5 (CO5)

PROGRAM NO: 1 **DATE: 17/01/2022**

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

PROGRAM

```
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")
f1.seek(0,0)
ff=f1.readlines()
for x in range(0,len(ff)):
    print(ff[x])
print()
print(ff)
f1.close()
```



OUTPUT

```
>>> %Run co5_1.py
 This is my first file in python.

 Want to work with files.

 This is my third line.

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

PROGRAM NO: 2 **DATE: 17/01/2022**

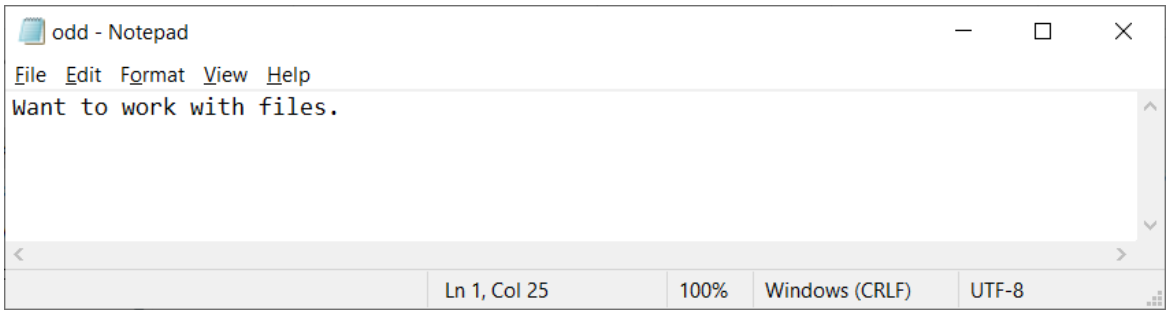
AIM : Write a program to copy odd lines of one file to other.

PROGRAM

```
f1=open("firstfile.txt","r")
for x in f1:
   print(x)

f1.seek(0,0)
print(".....\n")
f2=open("odd.txt","w")
ff=f1.readlines()


with open('odd.txt','w') as f2:
   for x in range(0,len(ff)):
      if(x%2!=0):
         print(ff[x])
         f2.write(ff[x])

print(".....\n")

f3=open("odd.txt","r")
for x in f3:
   print(x)
```

OUTPUT

```
>>> %Run co5_2.py
 This is my first file in python.

 Want to work with files.

 This is my third line.
 ........................

 Want to work with files.

 ........................

 Want to work with files.

>>>
```



PROGRAM NO: 3 **DATE: 31/01/2022**

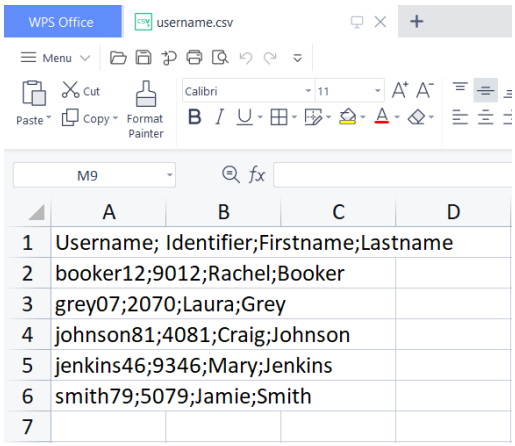
AIM : Write a Python program to read each row from a given csv file and print a list of strings.

PROGRAM

```
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)
print("..........................")

# 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 :\n")
for sl in rows:
  for l in sl:
      print(l)
      print()
cf.close()
```

username.csv



	A	B	C	D
1	Username; Identifier;Firstname;Lastname			
2	booker12;9012;Rachel;Booker			
3	grey07;2070;Laura;Grey			
4	johnson81;4081;Craig;Johnson			
5	jenkins46;9346;Mary;Jenkins			
6	smith79;5079;Jamie;Smith			
7				

OUTPUT

```
>>> %Run C05_3.py
 Username; Identifier;Firstname;Lastname

 ..............................
 [['booker12;9012;Rachel;Booker'], ['grey07;2070;Laura;Grey'], ['johnson
 81;4081;Craig;Johnson'], ['jenkins46;9346;Mary;Jenkins'], ['smith79;507
 9;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

>>>
```

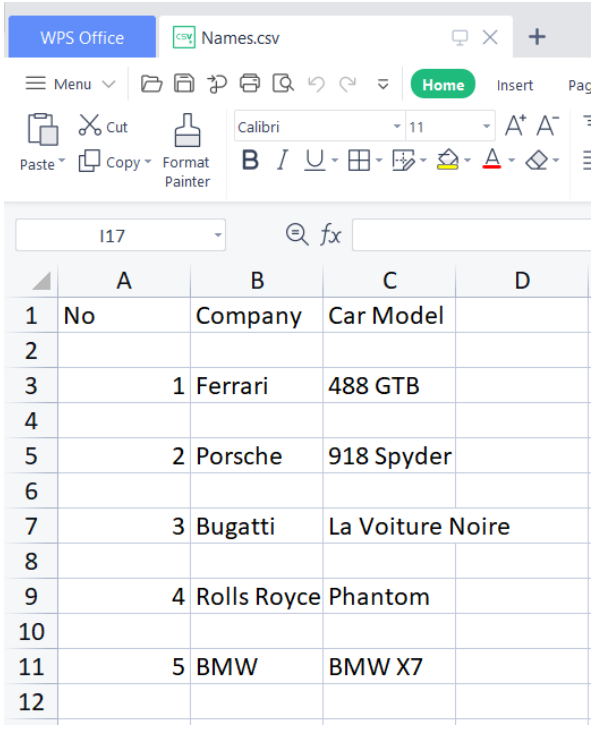
PROGRAM NO: 4 **DATE: 31/01/2022**

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

PROGRAM

```
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']))
```

Names.csv



	A	B	C	D
1	No	Company	Car Model	
2				
3	1	Ferrari	488 GTB	
4				
5	2	Porsche	918 Spyder	
6				
7	3	Bugatti	La Voiture Noire	
8				
9	4	Rolls Royce	Phantom	
10				
11	5	BMW	BMW X7	
12				

OUTPUT

```
>>> %Run CO5_4.py
 No Company
 1 Ferrari
 2 Porsche
 3 Bugatti
 4 Rolls Royce
 5 BMW

>>>
```

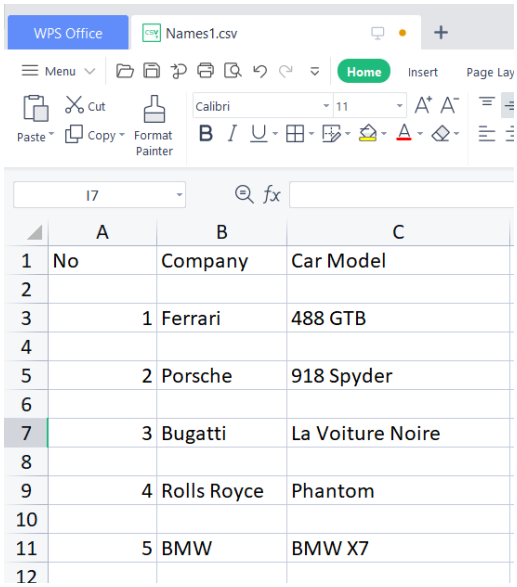
PROGRAM NO: 5 **DATE: 31/01/2022**

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.

PROGRAM

```
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)
```

Names.csv



	A	B	C
1	No	Company	Car Model
2			
3	1	Ferrari	488 GTB
4			
5	2	Porsche	918 Spyder
6			
7	3	Bugatti	La Voiture Noire
8			
9	4	Rolls Royce	Phantom
10			
11	5	BMW	BMW X7
12			

OUTPUT

```
>>> %Run co5_5.py
 No Company Car Model

 1 Ferrari 488 GTB
>>>
```