
PYTHON LAB

Course Outcome 1 (CO1):

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

CODE :

```
from datetime import datetime
current_year=datetime.today().year
final_year=int(input("Enter the last year"))
print("List of leap years")
for year in range(current_year,final_year):
    if(year%4==0)and(year%100!=0)or(year%400==0):
        print(year)
```

Output :

Enter the last year 2040

List of leap years

2024

2028

2032

2036

2. List comprehensions:

(a) Generate positive list of numbers from a given list of integers

CODE :

```
list1=[]
list2=[]
n1=int(input("Enter the number of elements"))
print("Enter integers")
for i in range(0,n1):
    a=int(input())
    list1.append(a)
print(list1)
print("List of positive numbers")
```

```
for i in list1:
    if(i>=0):
        list2.append(i)
print(list2)
```

Output :

Enter the number of elements 3

Enter integers

1

-2

0

[1, -2, 0]

List of positive numbers

[1, 0]

(b) Square of N numbers

CODE :

```
list1=[]
list2=[]
n1=int(input("Enter the number of elements"))
print("Enter integers")
for i in range(0,n1):
    a=int(input())
    list1.append(a)
print(list1)
print("Square of numbers")
for square in list1:
    square=square*square
    list2.append(square)
print(list2)
```

Output :

Enter the number of elements 5

Enter integers

1

2

3

-4

-5

[1, 2, 3, -4, -5]

Square of numbers

[1, 4, 9, 16, 25]

(c) Form a list of vowels selected from a given word

CODE

```
Word = input("Enter the word: ")
vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']
list=[]
for x in word:
    if(x in vowels and x not in list):
        list.append(x)
        print("vowels present in given word",list)
```

Output :

Enter the word: Reshu

Vowels present in given word ['e']

Vowels present in given word ['e', 'u']

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

CODE :

```
myinput=input("Message: ")
mylist=list(myinput)
for i in range(len(mylist)):
    mylist[i]=chr(ord(mylist[i])+1)
print(i+1)
```

Output :

Message: hello

5

3. Count the occurrences of each word in a line of text.

CODE :

```
def word_count(str):
    counts=dict()
    words=str.split()
    for word in words:
        if word in counts:
            counts[word]+= 1
        else:
            counts[word]= 1
    return counts
a=input("Enter the word: ")
print(word_count(a))
```

Output :

Enter the word: Gift of god

{'Gift': 1, 'of': 1, 'god': 1}

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

CODE :

```
lmt=int(input("Enter the limit"))
list1=[]
print("Enter the numbers")
for i in range(lmt):
    n=int(input())
    if(n>100):
        list1.append("over")
    else:
        list1.append(n)
print(list1)
```

Output :

Enter the limit 3

Enter the numbers

200

['over']

3

['over', 3]

4

['over', 3, 4]

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

CODE :

```
n1=int(input("Enter the limit : "))
list=[]
print("Enter the names")
for i in range(0,n1):
    f=input()
    list.append(f)
print("Entered first names")
print(list)
sum=0
for i in list:
    sum=sum+i.count("a")
print(sum)
```

Output :

Enter the limit : 3

Enter the names

alona

appu

ammu

Entered first names

['alona', 'appu', 'ammu']

4

6. 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

CODE :

```
lst1 = []
lst2 = []
```

```

lst1 = [(item) for item in input("Enter the list1 items
seperated by space: ").split()]

lst2 = [item for item in input("Enter the list2 items
seperated by space: ").split()]

sum1=str(0)
sum2=str(0)
if len(lst1)==len(lst2) :
    print(" Both list are of equal length")
else:
    print("Two list have unequal length")
for x in lst1:
    sum1=sum1+x
for x in lst2:
    sum2=sum2+x
if sum1==sum2:
    a="equal"
else:
    a="not equal"
print("Sum of two list are",a)
for x in lst1:
    for y in lst2:
        if x==y:
            print(y,"Occurs in both list")

```

Output :

Enter the list1 items seperated by space: 1 2 3 4

Enter the list2 items seperated by space: 3 5 2 100 12 12

Two list have unequal length

Sum of two list are not equal

2 Occurs in both list

3 Occurs in both list

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

CODE :

```
def change_char(str1):  
    char=str1[0]  
    str1=str1.replace(char, '$')  
    str1=char+str1[1:]  
    return str1  
a=input("Enter the string ")  
print(change_char(a))
```

Output :

Enter the string onion

oni\$n

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

CODE :

```
string=input("Enter a string ")  
new_str=string[-1]+string[1:-1]+string[0]  
print(new_str)
```

Output :

Enter a string teacher

reacht

9. Accept the radius from user and find area of circle.

CODE :

```
from math import pi  
r=float(input("Enter the radius "))  
calculate_Area=str(pi*r**2)  
print("The radius of the circle "+str(r)+" is"+calculate_Area)
```

Output :

Enter the radius 2

The radius of the circle 2.0 is 12.566370614359172

10. Find biggest of 3 numbers entered.

CODE :

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 > num2) and (num1 > num3):
    largest = num1
elif (num2 > num1) and (num2 > num3):
    largest = num2
else:
    largest = num3

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

Output :

Enter first number: 10

Enter second number: 9

Enter third number: 7

The largest number is 10.0

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

CODE :

```
filename = input("Input the Filename: ")
f_extns = filename.split(".")
print ("The extension of the file is : " + repr(f_extns[-1]))
```

Output :

Input the Filename: abc.jpeg

The extension of the file is : 'jpeg'

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

CODE :

```
list=[]
n=int(input("enter number of element in the list:"))
print("enter colors of the list")
for i in range(0,n):
    a=input()
    list.append(a)
print("list is:",list)
print("first color in the list is:",list[0])
print("last color in the list is:",list[-1])
```

Output:

enter number of element in the list:3

enter colors of the list

red

black

blue

list is: ['red', 'black', 'blue']

first color in the list is: red

last color in the list is: blue

13. Accept an integer n and compute $n+nn+nnn$.

CODE :

```
n=int(input("Enter a number :"))
value=n+n*n+n*n*n
print("value of n+n*n+n*n*n is =",value)
```

Output :

Enter a number :5

value of n+n*n+n*n*n is = 155

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

CODE :

```
list1=[]
n1=int(input("enter number of elements in list 1:"))
print("enter colors")
for i in range(0,n1):
    a=input()
    list1.append(a)
print(list1)
list2=[]
n2=int(input("enter number of elements in list 2:"))
print("enter colors")
for i in range(0,n2):
    b=input()
    list2.append(b)
print(list2)
```

Output :

enter number of elements in list 1:2

enter colors

red

black

['red', 'black']

enter number of elements in list 2:2

enter colors

red

yellow

['red ', 'yellow']

- 15.** Create a single string separated with space from two strings by swapping the character at position 1.

CODE :

```
def chars_mix_up(a, b):  
    new_a = b[:1] + a[1:]  
    new_b = a[:1] + b[1:]  
  
    return new_a + ' ' + new_b  
a=input("Enter the string1 ")  
b=input("Enter the string2 ")  
print(chars_mix_up(a,b))
```

Output :

Enter the string1 abc

Enter the string2 xyz

xbc ayz

- 16.** Sort dictionary in ascending and descending order.

CODE :

```
y={'carl':40,'alan':2,'bob':1,'danny':3}  
l=list(y.items())  
l.sort()  
print("Ascending order is",l)  
l=list(y.items())  
l.sort(reverse=True)
```

```
print("Descending order is",l)
dict=dict(l)
print("Dictionary",dict)
```

Output :

Ascending order is [('alan', 2), ('bob', 1), ('carl', 40), ('danny', 3)]

Descending order is [('danny', 3), ('carl', 40), ('bob', 1), ('alan', 2)]

Dictionary {'danny': 3, 'carl': 40, 'bob': 1, 'alan': 2}

17. Merge two dictionaries.

CODE :

```
d1={}
n1=int(input("enter limit in 1st dictionary"))
print("enter dictionary value")
for i in range(0,n1):
    key=input("key:")
    value=input("values:")
    d1.update({key:value})
d2={}
n2=int(input("enter limit in 1st dictionary"))
print("enter dictionary value")
for i in range(0,n2):
    key=input("key:")
    value=input("values:")
    d2.update({key:value})
print("first dictionary:",d1)
print("second dictionary:",d2)
d3={}
for i in (d1,d2):
    d3.update(i)
print("merged dictionary is",d3)
```

Output :

enter limit in 1st dictionary2

enter dictionary value

key:e

values:3

key:g

values:5

enter limit in 1st dictionary2

enter dictionary value

key:n

values:7

key:m

values:8

first dictionary: {'e': '3', 'g': '5'}

second dictionary: {'n': '7', 'm': '8'}

merged dictionary is {'e': '3', 'g': '5', 'n': '7', 'm': '8'}

18. Find gcd of 2 numbers.

CODE :

```
def gcd_fun (x, y):  
    if (y == 0):  
        return x  
    else:  
        return gcd_fun (y, x % y)  
x =int (input ("Enter the first number: "))  
y =int (input ("Enter the second number: "))  
num = gcd_fun(x, y)  
print("GCD of two number is: ")  
print(num)
```

Output :

Enter the first number: 60

Enter the second number: 48

GCD of two number is:

12

- 19.** From a list of integers, create a list removing even numbers.

CODE :

```
list1=[]
n1=int(input("Enter the number of elements"))
print("Enter integers")
for i in range(0,n1):
    a=int(input())
    list1.append(a)
print("List of after removing even numbers")
for i in list1:
    if(i%2==0):
        list1.remove(i)
print(list1)
```

Output :

Enter the number of elements5

Enter integers

11

22

33

44

55

List of after removing even numbers

[11, 33, 55]

Course Outcome 2 (CO2)

1. Program to find the factorial of a number

CODE :

```
num = int(input("Enter a number: "))
factorial = 1
if num < 0:
    print(" Factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Output :

Enter a number: 3

The factorial of 3 is 6

2. Generate Fibonacci series of N terms

CODE :

```
n = int(input("Enter the value of 'n': "))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci Series: ", end = " ")
while(count <= n):
    print(sum, end = " ")
    count += 1
    a = b
    b = sum
    sum = a + b
```

Output :

Enter the value of 'n': 3

Fibonacci Series: 0 1 1

3. Find the sum of all items in a list

CODE:

```
lst = []
num = int(input('How many numbers: '))
for n in range(num):
    numbers = int(input('Enter number '))
    lst.append(numbers)
print("Sum of elements in given list is :", sum(lst))
```

Output :

How many numbers: 3

Enter number 4

Enter number 5

Enter number 6

Sum of elements in given list is : 15

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

CODE:

```
n1=int(input("Enter the first number"))
n2=int(input("Enter the last number"))
for i in range(n1,n2):
    for j in range(32,100,1):
        if i==j*j:
            string=str(i)
```

```
        if int(string[0])%2==0 and int(string[1])%2==0
        and int(string[2])%2==0 and int(string[3])%2==0:
            print(i)
```

Output:

Enter the first number1000

Enter the last number9999

4624

6084

6400

8464

5. Display the given pyramid with step number accepted from user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

CODE :

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

Output :

Enter the step number4

1

2 4

3 6 9

4 8 12 16

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

CODE :

```
def char_frequency(str1):  
    dict = {}  
    for n in str1:  
        keys = dict.keys()  
        if n in keys:  
            dict[n] += 1  
        else:  
            dict[n] = 1  
    return dict  
a=input("Enter the string ")  
print(char_frequency(a))
```

Output :

Enter the string apple

{'a': 1, 'p': 2, 'l': 1, 'e': 1}

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

CODE:

```
string = input("Enter a string ")  
if len(string) < 3:  
    print(string)  
elif string[-3:] == 'ing':  
    print(string + 'ly')  
else:  
    print(string + 'ing')
```

Output :

Enter a string abc

abcing

8. Accept a list of words and return length of longest word.

CODE:

```
def longestWordLength(string):  
    length=0  
    w= ''
```

```

    for word in string.split():
        if(len(word)>length):
            length=len(word)
            w=word
    return(length,w)
string=input("Enter the string")
l,w=longestWordLength(string)
print("Longest word is",w, "and its length is",l)

```

Output :

Enter the string three and four is

Longest word is three and its length is 5

9. Construct following pattern using nested loop

```

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*

```

CODE:

```

n=int(input("Enter tne number "))
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 number 5

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

10. Generate all factors of a number.

CODE:

```
def print_factors(x):  
    print("The factors of",x,"are:")  
    for i in range(1, x + 1):  
        if x % i == 0:  
            print(i)  
num = int(input("Enter the number "))  
print_factors(num)
```

Output :

Enter the number 65

The factors of 65 are:

1

5

13

65

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

CODE:

```
import math
a=int(input("Enter the value"))
s_area=lambda a:a*a
print("Area of square is : ",s_area(a))
b=int(input("Enter the length"))
c=int(input("Enter the height"))
r_area=lambda len,ht:len*ht
print("Area of rectangle is : ",r_area(b,c))
d=int(input("Enter the base"))
e=int(input("Enter the height"))
t_area=lambda b,h:0.5*b*h
print("Area of triangle is : ",r_area(d,e))
```

Output :

Enter the value2

Area of square is : 4

Enter the length3

Enter the height4

Area of rectangle is : 12

Enter the base3

Enter the height2

Area of triangle is : 3

Course Outcome 3(CO3):

1. Work with built-in packages

CODE:

```
import math
print(math.sqrt(25))
print(math.pi)
print(math.degrees(2))
print(math.radians(60))
print(math.sin(2))
print(math.cos(0.5))
print(math.tan(0.23))
print(math.factorial(4))
import random
print(random.randint(0, 5))
print(random.random())
print(random.random() * 100)
List = [1, 4, True, 800, "python", 27, "hello"]
print(random.choice(List))
import datetime
from datetime import date
import time
print(time.time())
print(date.fromtimestamp(454554))
```

Output :

5.0

3.141592653589793

114.59155902616465

1.0471975511965976

0.9092974268256817

0.8775825618903728

0.23414336235146527

24

1

0.03772466762864812

24.55738973412146

27

1645957476.892115

1970-01-06

2. 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)

Course Outcome 4 (CO4):

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

CODE:

```

class Rectangle1:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area1(self):
        return self.length * self.breadth
a = int(input("Enter length of 1st rectangle: "))
b = int(input("Enter breadth of 1st rectangle: "))
r1= Rectangle1(a,b)
print("Area of 1st rectangle:", r1.area1())

class Rectangle2:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area2(self):
        return self.length * self.breadth
c= int(input("Enter length of 2nd rectangle: "))
d = int(input("Enter breadth of 2nd rectangle: "))
r2 = Rectangle2(c,d)
print("Area of 2nd rectangle:", r2.area2())

if (r1.area1()>r2.area2()):
    print(" 1st rectangle large")
else:
    print(" 2nd rectangle large")

```

Output :

Enter length of 1st rectangle: 2

Enter breadth of 1st rectangle: 3

Area of 1st rectangle: 6

Enter length of 2nd rectangle: 4

Enter breadth of 2nd rectangle: 3

Area of 2nd rectangle: 12

2nd rectangle large

2. 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.

CODE:

```
class Bank_Account:
    def __init__(self):
        self.balance = 0
        print("Hello!!! Welcome to the Deposit & Withdrawal Machine")

    def deposit(self):
        amount = float(input("Enter amount to be Deposited:
"))
        self.balance += amount
        print("\n Amount Deposited:", amount)

    def withdraw(self):
        amount = float(input("Enter amount to be Withdrawn:
"))
        if self.balance >= amount:
            self.balance -= amount
            print("\n You Withdrew:", amount)
        else:
            print("\n Insufficient balance ")

    def display(self):
        print("\n Net Available Balance=", self.balance)
s = Bank_Account()
s.deposit()
s.withdraw()
s.display()
```

Output :

Hello!!! Welcome to the Deposit & Withdrawal Machine

Enter amount to be Deposited: 2000

Amount Deposited: 2000.0

Enter amount to be Withdrawn: 200

You Withdrew: 200.0

Net Available Balance= 1800.0

3. Create a class Rectangle with private attributes length and width. Overload ‘.

CODE:

```
class Rectangle:
    def __init__(self, length,breadth):
        self.length = length
        self.breadth = breadth
    def cal_area(self):
        self.area=self.length * self.breadth
        print('Area : ',self.area)
    def __lt__(self, second):
        if self.area < second.area:
            return True
        else:
            return False

print('Enter length and breadth of rectangle 1:')
l,b=int(input()),int(input())
print('Enter length and breadth of rectangle 2:')
l2,b2=int(input()),int(input())
print('rectangle 1 area:')
r = Rectangle(l,b)
r.cal_area()
print('rectangle 2 area:')
r2 = Rectangle(l2,b2)
r2.cal_area()
if r < r2:
    print("\nRectangle two is large")
else:
    print("Rectangle one is large or these are equal")
```

Output :

Enter length and breadth of rectangle 1:

```
2
3
Enter length and breadth of rectangle 2:
3
4
rectangle 1 area:
Area : 6
rectangle 2 area:
Area : 12
```

Rectangle two is large

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

CODE:

```
class Time:
    def __init__(self, hour, minute, second):
        self.hour=hour
        self.minute=minute
        self.second=second

    def cal_time(self):
        self.sum = self.hour + self.minute + self.second
        print('Time : ', self.sum)

    def __add__(self, second):
        if self.sum + second.sum:
            return True
        else:
            return False

print('Enter hour,minute and second of clock 1:')
h,m,s = int(input()), int(input()) , int(input())
print('Enter hour,minute and second of clock 2:')
h1,m1,s1 = int(input()), int(input()) , int(input())
print('clock 1 sum:')
t = Time(h,m,s)
t.cal_time()
print('clock 2 sum:')
t2 = Time(h1, m1,s1)
t2.cal_time()
```

Output :

Enter hour,minute and second of clock 1:

2

30

56

Enter hour,minute and second of clock 2:

3

12

78

clock 1 sum:

Time : 88

clock 2 sum:

Time : 93

5. 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

CODE:

```
class Publisher:
    def read(self):
        print("Books")
class Book(Publisher):
    def title(self):
        print("Title : Python Programming")
    def author(self):
        print("Author : Ryan Tuner")
class Python(Book):
    def price(self):
```

```
        print("Price : 2324/-")
    def pages(self):
        print("Pages : 274")
p=Python()
p.read()
p.title()
p.author()
p.price()
p.pages()
```

Output :

Books
Title : Python Programming
Author : Ryan Tuner
Price : 2324/-
Pages : 274

Course Outcome 5 (CO5):

1. Write a Python program to read a file line by line and store it into a list.

CODE:

```
with open("test.txt") as f:
    content_list = f.readlines()
content_list = [x.strip() for x in content_list]
print(content_list)
```

test.txt

word
excel
spreadsheet

Output :

['word', 'excel', 'spreadsheet']

2. Python program to copy odd lines of one file to other

CODE:

```
fn = open('test.txt', 'r')
fn1 = open('test2.txt', 'w')
cont = fn.readlines()
type(cont)
for i in range(0, len(cont)):
    if (i % 2 != 0):
        fn1.write(cont[i])
    else:
        pass
fn1.close()

fn1 = open('test2.txt', 'r')
cont1 = fn1.read()
print(cont1)
fn.close()
fn1.close()
```

test2.txt

excel

Output :

Excel

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

CODE:

```
import csv
with open('people.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

people.csv

Sn,Name,Age,Job
1,Alona,26,Teacher


```
2,Teres,31,Engineer
3,Jiby,32,Doctor
```

Output :

```
['Sn', 'Name', 'Age', 'Job']
['1', 'Alona', '26', 'Teacher']
['2', 'Teres', '31', 'Engineer']
['3', 'Jiby', '32', 'Doctor']
```

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

CODE:

```
import csv
with open('people.csv') as csvfile:
    data = csv.DictReader(csvfile)
    print("Employee name")
    print("-----")
    for row in data:
        print(row['Name'], row['Age'])
```

Output :

```
Employee name
-----
Alona 26
Teres 31
Jiby 32
```

5. 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.

CODE:

```

import csv
csv_columns = ['id', 'Column1', 'Column2', 'Column3',
               'Column4', 'Column5']
dict_data = {'id': ['1', '2', '3'],
             'Column1': [33, 25, 56],
             'Column2': [35, 30, 30],
             'Column3': [21, 40, 55],
             'Column4': [71, 25, 55],
             'Column5': [10, 10, 40], }
csv_file = "temp.csv"
try:
    with open(csv_file, 'w') as csvfile:
        writer = csv.DictWriter(csvfile,
                                fieldnames=csv_columns)
        writer.writeheader()
        for data in dict_data:
            writer.writerow(dict_data)
except IOError:
    print("I/O error")
data = csv.DictReader(open(csv_file))
print("CSV file as a dictionary:\n")
for row in data:
    print(row)

```

temp.csv

```

id,Column1,Column2,Column3,Column4,Column5

["1", "2", "3"], [33, 25, 56], [35, 30, 30], [21, 40, 55], [71, 25, 55], [10, 10, 40]

["1", "2", "3"], [33, 25, 56], [35, 30, 30], [21, 40, 55], [71, 25, 55], [10, 10, 40]

["1", "2", "3"], [33, 25, 56], [35, 30, 30], [21, 40, 55], [71, 25, 55], [10, 10, 40]

["1", "2", "3"], [33, 25, 56], [35, 30, 30], [21, 40, 55], [71, 25, 55], [10, 10, 40]

["1", "2", "3"], [33, 25, 56], [35, 30, 30], [21, 40, 55], [71, 25, 55], [10, 10, 40]

```

Output :

CSV file as a dictionary:

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'),  
('Column4', '[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'),  
('Column4', '[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'),  
('Column4', '[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'),  
('Column4', '[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'), ('Column4',  
'[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```

```
OrderedDict([('id', "[1', '2', '3']"), ('Column1', '[33, 25, 56]'),  
('Column2', '[35, 30, 30]'), ('Column3', '[21, 40, 55]'), ('Column4',  
'[71, 25, 55]'), ('Column5', '[10, 10, 40]')])
```