Experiment No.: 1

Aim

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

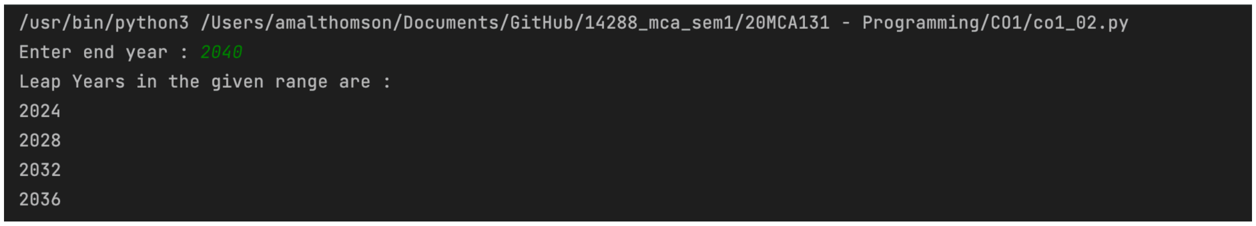
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
end = int(input("Enter end year : "))
print("Leap Years in the given range are : ")
for year in range(2022, end):
  if (0 == year % 4) and (0 != year % 100):
    print (year)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_02.py
Enter end year : 2040
Leap Years in the given range are :
2024
2028
2032
2036
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 2

Aim

List comprehensions:
(a) Generate positive list of numbers from a given list of integers
(b) Square of N numbers
(c) Form a list of vowels selected from a given word.

CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

(a)

```
x=int(input("Enter the size of list 1 : "))

lx=[]

for i in range (0,x):

   a=int(input("Enter Number : "))

   lx.append(a)

print("\nList 1 is ;", lx)

p=[i for i in lx if i>0]

print("\nList of Positive numbers is :", p)
```

(b)

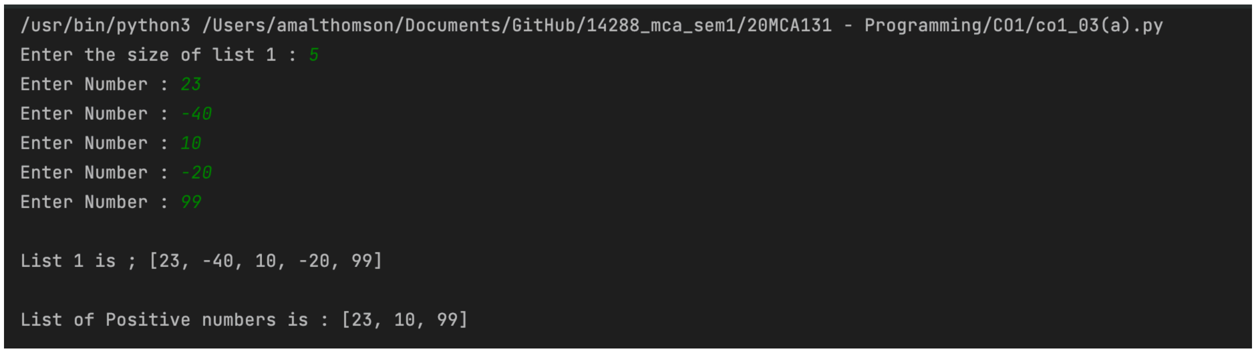
```
x=int(input("Enter the lower limit : "))

y=int(input("Enter the upper limit : "))

sq=[i*i for i in range(x, y)]

print(sq)
```

(c)

```
word=str(input("Enter a WORD : "))
vow=[x for x in word if x=='a' or x=='e' or x=='i'
    or x=="o" or x=='u' or x=="A" or x=="E" or x=="I" or x=="O" or x=='U']
print(vow)
```

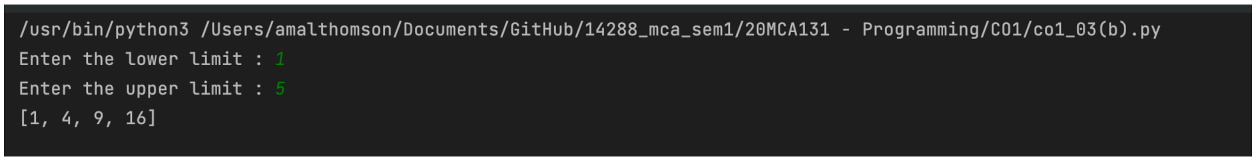
Output Screenshot

(a)



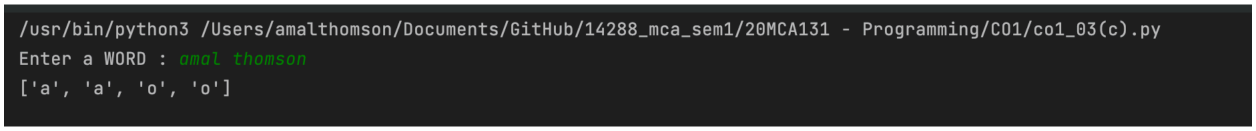
```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_03(a).py
Enter the size of list 1 : 5
Enter Number : 23
Enter Number : -40
Enter Number : 10
Enter Number : -20
Enter Number : 99

List 1 is ; [23, -40, 10, -20, 99]

List of Positive numbers is : [23, 10, 99]
```

(b)



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_03(b).py
Enter the lower limit : 1
Enter the upper limit : 5
[1, 4, 9, 16]
```

(c)



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_03(c).py
Enter a WORD : amal thomson
['a', 'a', 'o', 'o']
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 3

Aim

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

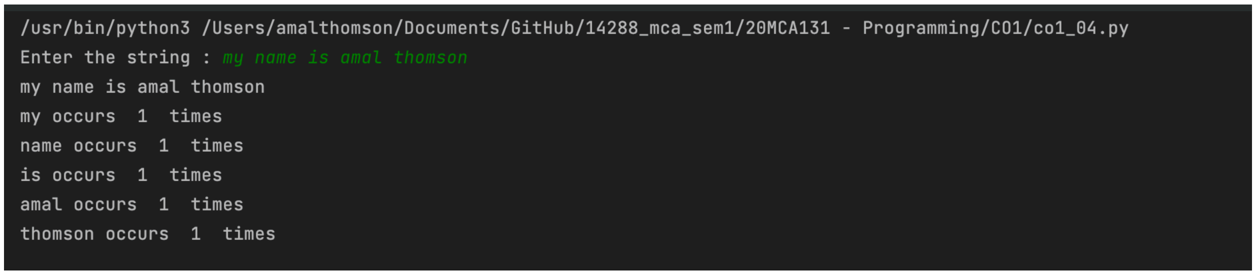
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
string=str(input("Enter the string : "))
a={}
print(string)
s=string.split()
for i in s:
    if i in a:
        a[i]=a[i]+1
    else:
        a[i]=1
for m,n in a.items():
    print(m,"occurs ",n," times")
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_04.py
Enter the string : my name is amal thomson
my name is amal thomson
my occurs  1  times
name occurs  1  times
is occurs  1  times
amal occurs  1  times
thomson occurs  1  times
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 4

Aim

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

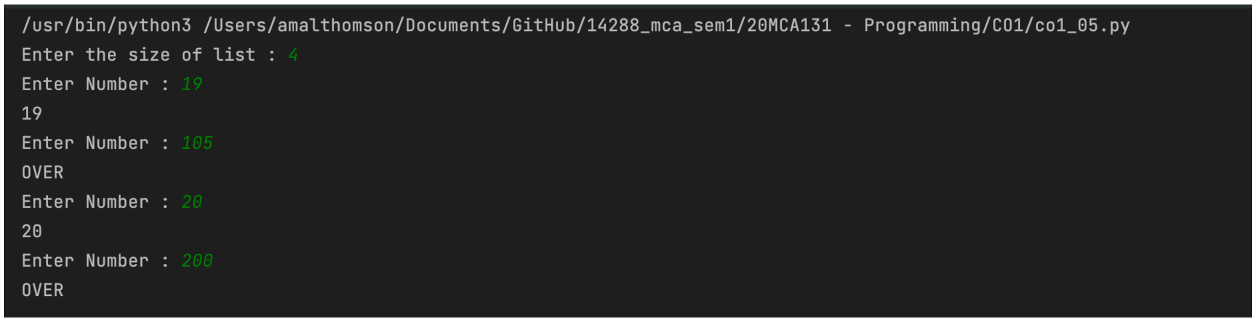
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
for i in range(0, x):
    a=int(input("Enter Number : "))
    if (a>100):
        a="OVER"
        print(a)
    else:
        print(a)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_05.py
Enter the size of list : 4
Enter Number : 19
19
Enter Number : 105
OVER
Enter Number : 20
20
Enter Number : 200
OVER
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 5

Aim

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

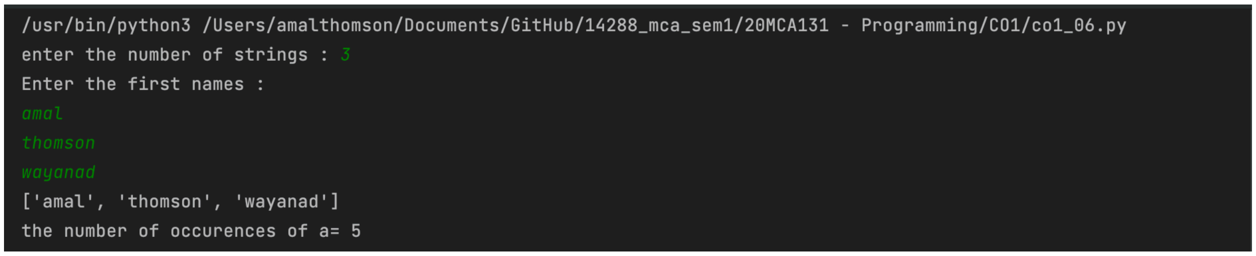
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
x=int(input("enter the number of strings"))
a=[]
flag=0
print("Enter the first name")
for i in range(0,x):
    s=input()
    a.append(s)
print(a)
for i in a:
    for j in i:
        if(j=="a"):
            flag=flag+1
print("the number of occurences of a=",flag)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_06.py
enter the number of strings : 3
Enter the first names :
amal
thomson
wayanad
['amal', 'thomson', 'wayanad']
the number of occurences of a= 5
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 6

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

CO1

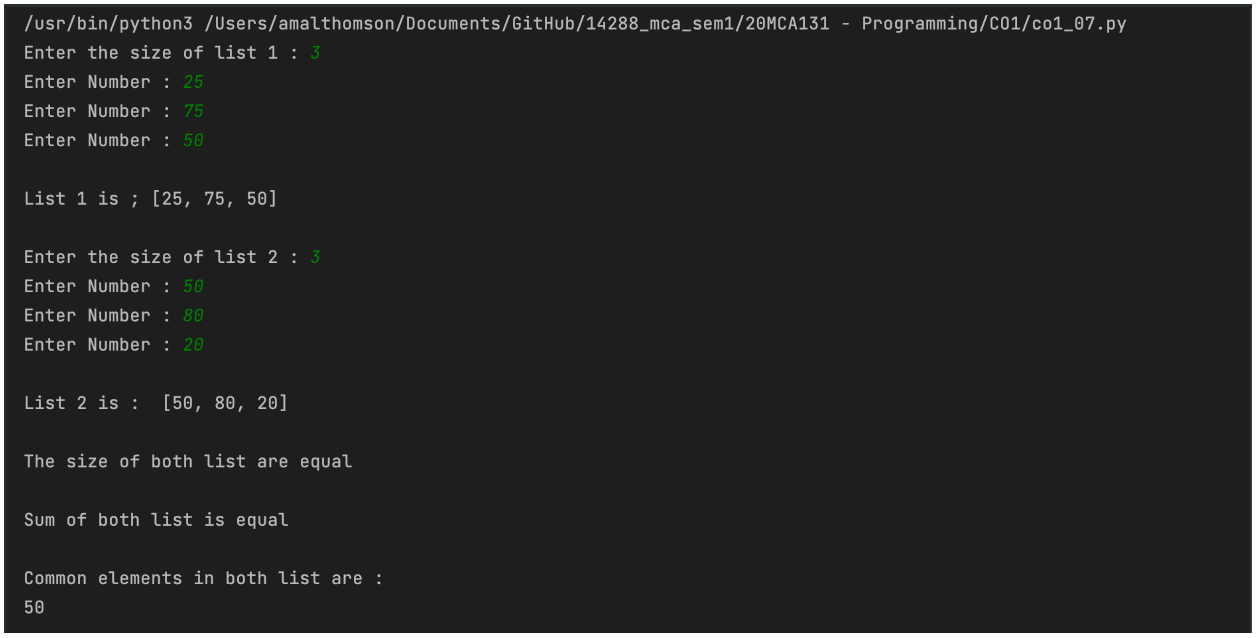
Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
x=int(input("Enter the size of list 1 : "))
lx=[]
for i in range (0,x):
   a=int(input("Enter Number : "))
   lx.append(a)
print("\nList 1 is ;", lx)
y=int(input("\nEnter the size of list 2 : "))
ly=[]
for i in range (0,y):
   b=int(input("Enter Number : "))
   ly.append(b)
print("\nList 2 is : ", ly)
if (len(lx)==len(ly)):
   print("\nThe size of both list are equal")
else:
   print("\nThe size of both list are not equal")
```

```
if (sum(lx)==sum(ly)):
   print("\nSum of both list is equal")
else:
   print("\nSum of both list are not equal")
print("\nCommon elements in both list are : ")
for i in lx:
   for j in ly:
      if(i==j):
         print(i)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_07.py
Enter the size of list 1 : 3
Enter Number : 25
Enter Number : 75
Enter Number : 50

List 1 is ; [25, 75, 50]

Enter the size of list 2 : 3
Enter Number : 50
Enter Number : 80
Enter Number : 20

List 2 is :  [50, 80, 20]

The size of both list are equal

Sum of both list is equal

Common elements in both list are :
50
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 7

Aim

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

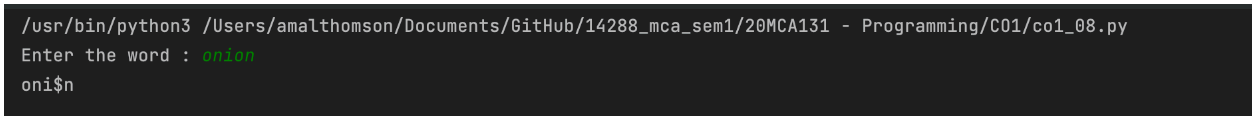
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
word=str(input("Enter the word : "))
a=word[0]
for i in word:
   if (i==a):
      word=word.replace(i, "$")
      word=a+word[1:]
print(word)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_08.py
Enter the word : onion
oni$n
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 8

Aim

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

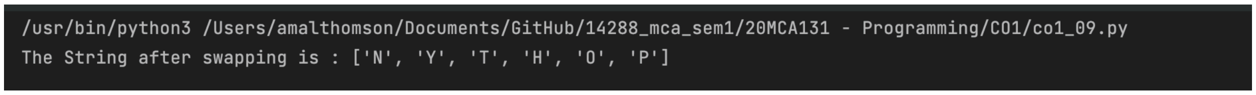
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
string=["P","Y", "T", "H", "O", "N"]

string[0]="N"

string[5]="P"

print("The String after swapping is :", string)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_09.py
The String after swapping is : ['N', 'Y', 'T', 'H', 'O', 'P']
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 9

Aim

Accept the radius from user and find area of circle.

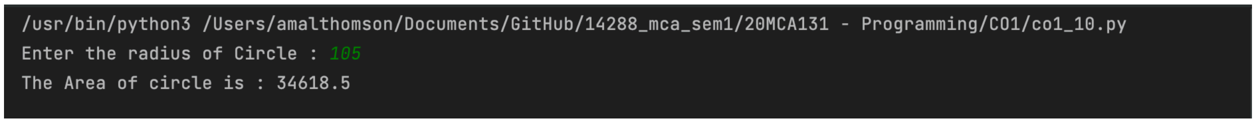
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
radius=int(input("Enter the radius of Circle :"))

area=3.14*radius*radius

print("The Area of circle is :", area)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_10.py
Enter the radius of Circle : 105
The Area of circle is : 34618.5
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 10

Aim

Find biggest of 3 numbers entered.

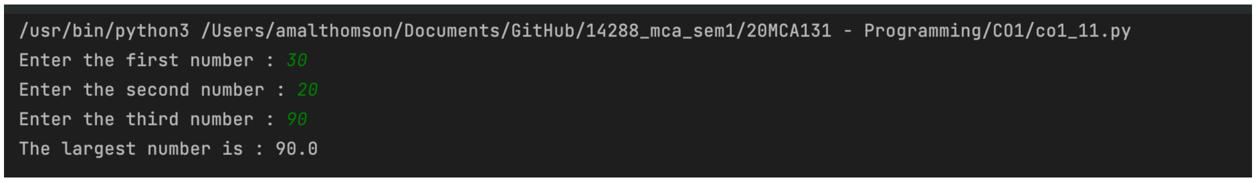
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
a=float(input("Enter the first number : "))
b=float(input("Enter the second number : "))
c=float(input("Enter the third number : "))
m=max(a, b, c)
print("The largest number is :", m)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_11.py
Enter the first number : 30
Enter the second number : 20
Enter the third number : 90
The largest number is : 90.0
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 11

Aim

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

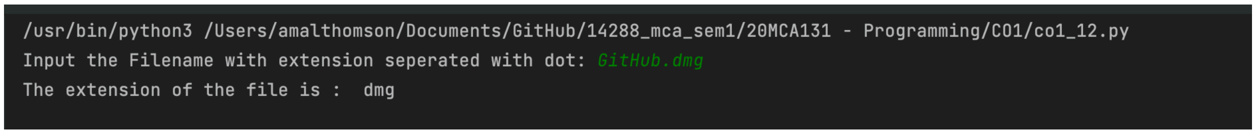
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
filename = input("Input the Filename with extension seperated with dot: ")
extension = filename.split(".")
print ("The extension of the file is : ", extension[-1])
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_12.py
Input the Filename with extension seperated with dot: GitHub.dmg
The extension of the file is :  dmg
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 12

Aim

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

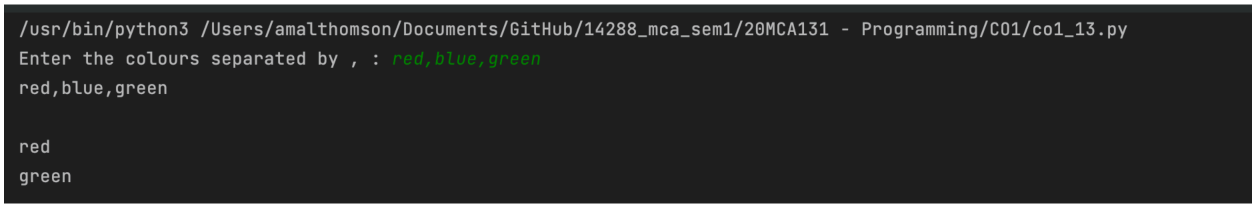
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
colour=input("Enter the colours seperated by , ")

print(colour, "\n")

x=colour.split(",")

print(x[0])

print(x[-1])
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_13.py
Enter the colours separated by , : red,blue,green
red,blue,green

red
green
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 13

Aim

Accept an integer n and compute n+nn+nnn

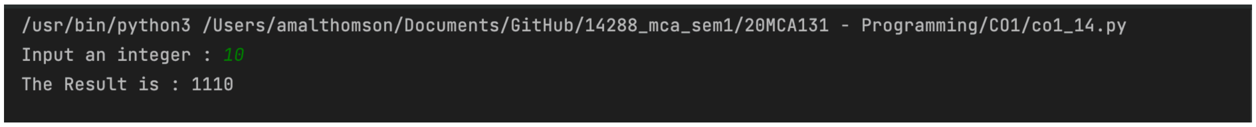
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
a = int(input("Input an integer : "))
x=a
y=a*a
z=a*a*a
s=x+y+z
print("The Result is :", s)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_14.py
Input an integer : 10
The Result is : 1110
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 14

Aim

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

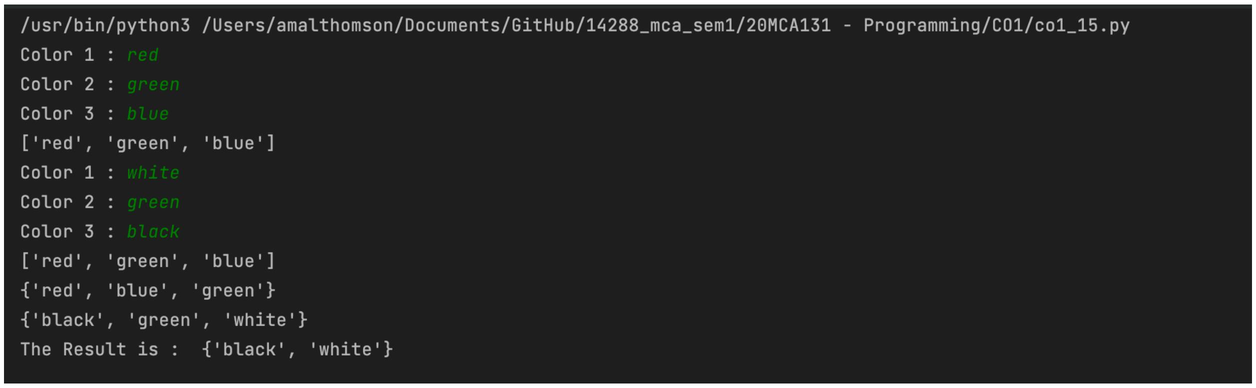
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
a=str(input("Color 1 : "))
b=str(input("Color 2 : "))
c=str(input("Color 3 : "))
x=[a, b, c]
print(x)
d=str(input("Color 1 : "))
e=str(input("Color 2 : "))
f=str(input("Color 3 : "))
y=[d, e, f]
print(x)
ax=set(x)
ay=set(y)
print(ax)
print(ay)
az=ay.difference(ax)
print("The Result is : ", az)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_15.py
Color 1 : red
Color 2 : green
Color 3 : blue
['red', 'green', 'blue']
Color 1 : white
Color 2 : green
Color 3 : black
['red', 'green', 'blue']
{'red', 'blue', 'green'}
{'black', 'green', 'white'}
The Result is :  {'black', 'white'}
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 15

Aim

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

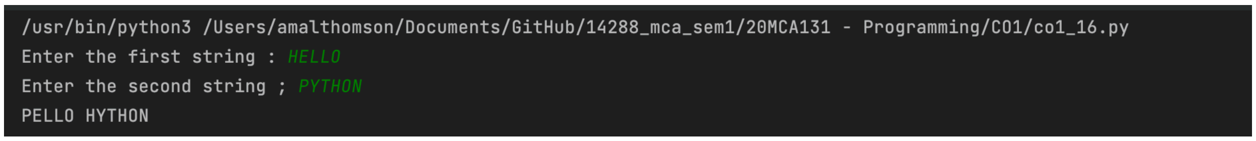
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
x=input("Enter the first string : ")

y=input("Enter the second string ; ")

a=x[0]

b=y[0]

x1=y[0]+x[1:]

y1=x[0]+y[1:]

z=x1+" "+y1

print(z)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_16.py
Enter the first string : HELLO
Enter the second string ; PYTHON
PELLO HYTHON
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 16

Aim

Sort dictionary in ascending and descending order.

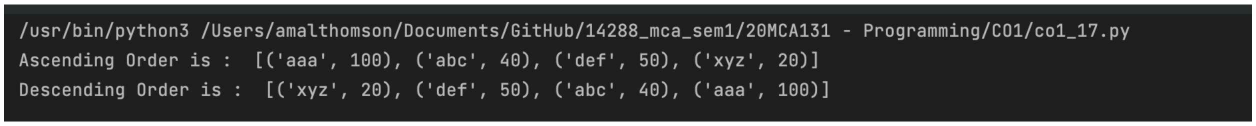
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
y={'abc':40, 'xyz':20, 'def':50, 'aaa':100}
a=list(y.items())
a.sort()
print("Ascending Order is : ", a)
b=list(y.items())
b.sort(reverse=True)
print("Descending Order is : ", b)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_17.py
Ascending Order is :  [('aaa', 100), ('abc', 40), ('def', 50), ('xyz', 20)]
Descending Order is :  [('xyz', 20), ('def', 50), ('abc', 40), ('aaa', 100)]
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 17

Aim

Merge two dictionaries.

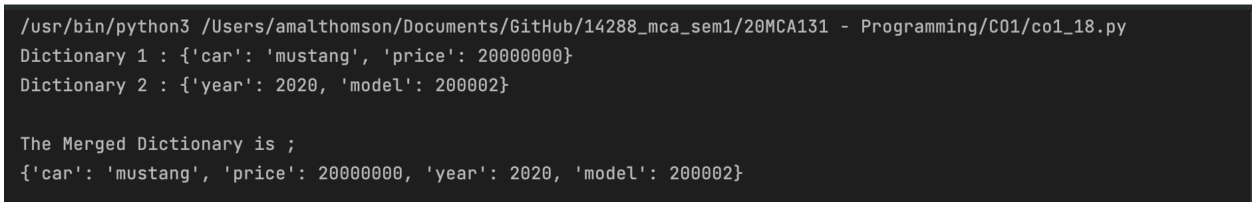
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
dic1={'car':'mustang', 'price':20000000}
dic2={'year':2020, 'model':200002}
print("Dictionary 1 :", dic1)
print("Dictionary 2 :", dic2)
d=dic1.copy()
d.update(dic2)
print("\nThe Merged Dictionary is ; ")
print(d)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_18.py
Dictionary 1 : {'car': 'mustang', 'price': 20000000}
Dictionary 2 : {'year': 2020, 'model': 200002}

The Merged Dictionary is ;
{'car': 'mustang', 'price': 20000000, 'year': 2020, 'model': 200002}
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 18

Aim

Find gcd of 2 numbers.

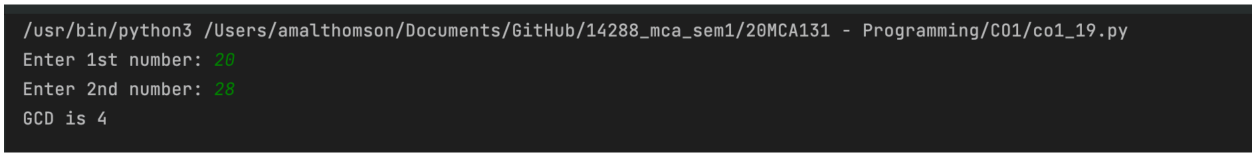
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
num1=int(input("Enter 1st number: "))
num2=int(input("Enter 2nd number: "))
i = 1
while(i<=num1 and i<=num2):
  if(num1%i==0 and num2%i==0):
    gcd = i
  i = i + 1
print("GCD is", gcd)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_19.py
Enter 1st number: 20
Enter 2nd number: 28
GCD is 4
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 19

Aim

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

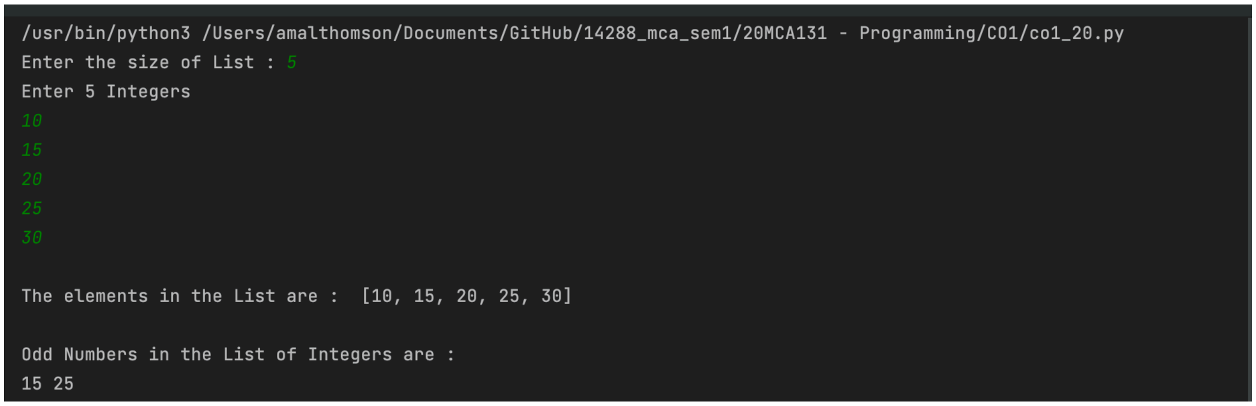
CO1

Understand the basics of python programming language including input/output functions, operators, basics and collection datatypes.

Procedure

```
integers = []
n = int(input("Enter the size of List : "))
print("Enter", n, "Integers")
for i in range(0, n):
    elements = int(input())
    integers.append(elements)
print("\nThe elements in the List are : ", integers)
print("\nOdd Numbers in the List of Integers are : ")
for num in integers:
    if num % 2 != 0:
        print(num, end=" ")
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO1/co1_20.py
Enter the size of List : 5
Enter 5 Integers
10
15
20
25
30

The elements in the List are :  [10, 15, 20, 25, 30]

Odd Numbers in the List of Integers are :
15 25
```

Result

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

Experiment No.: 20

Aim

Program to find the factorial of a number

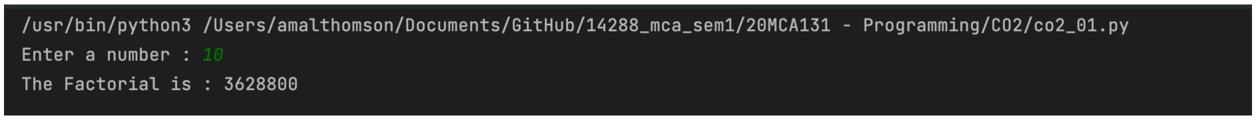
CO2

Implement decision making, looping constructs and functions

Procedure

```
num=int(input("Enter a number : "))
fac=1
while (1<=num):
    fac=fac*num
    num=num-1
print("The Factorial is :", fac)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_01.py
Enter a number : 10
The Factorial is : 3628800
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 21

Aim

Generate Fibonacci series of N terms

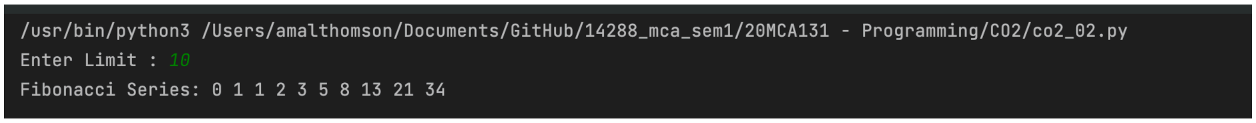
CO2

Implement decision making, looping constructs and functions

Procedure

```
num = int(input("Enter Limit : "))
n1 = 0
n2 = 1
print("Fibonacci Series:", n1, n2, end=" ")
for i in range(2, num):
    n3 = n1 + n2
    n1 = n2
    n2 = n3
    print(n3, end=" ")
print()
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_02.py
Enter Limit : 10
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 22

Aim

Find the sum of all items in a list

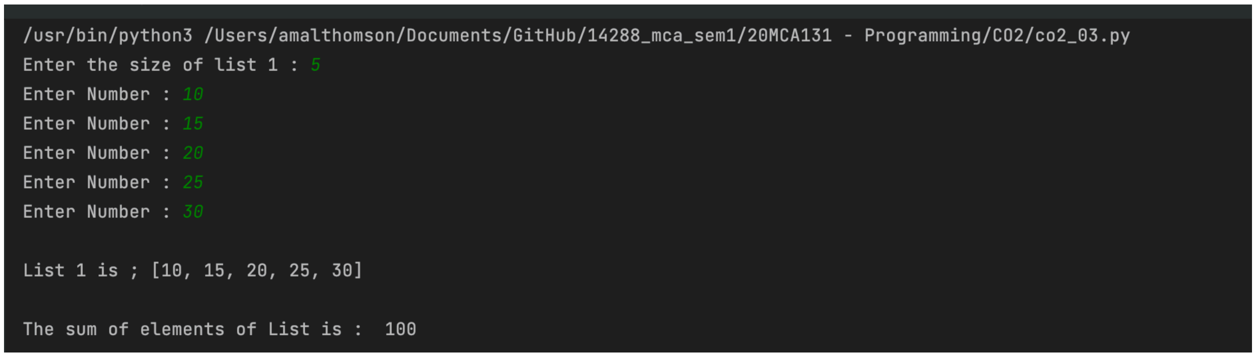
CO2

Implement decision making, looping constructs and functions

Procedure

```
x=int(input("Enter the size of list 1 : "))
lx=[]
for i in range (0,x):
    a=int(input("Enter Number : "))
    lx.append(a)
print("\nList 1 is ;", lx)
print("\nThe of elements of List is : ", sum(lx))
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_03.py
Enter the size of list 1 : 5
Enter Number : 10
Enter Number : 15
Enter Number : 20
Enter Number : 25
Enter Number : 30

List 1 is ; [10, 15, 20, 25, 30]

The sum of elements of List is :  100
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 23

Aim

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

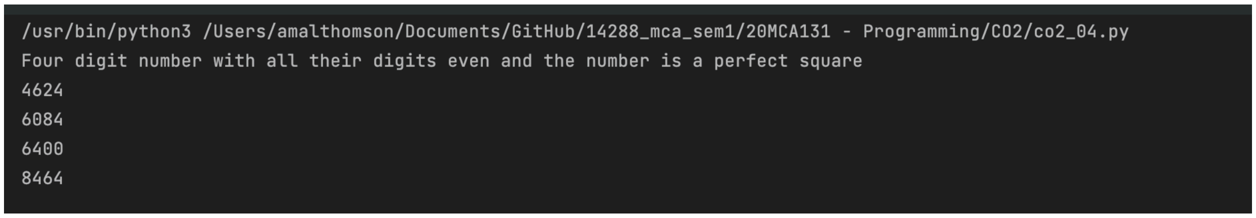
CO2

Implement decision making, looping constructs and functions

Procedure

```
print("Four digit number with all their digits even and the number is a perfect square")
for i in range(1000,10000):
    for j in range(32,100):
        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 Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_04.py
Four digit number with all their digits even and the number is a perfect square
4624
6084
6400
8464
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 24

Aim

Display the given pyramid with step number accepted from user. Eg: N=4

```
1
2 4
3 6 9
4 8 12 16
```

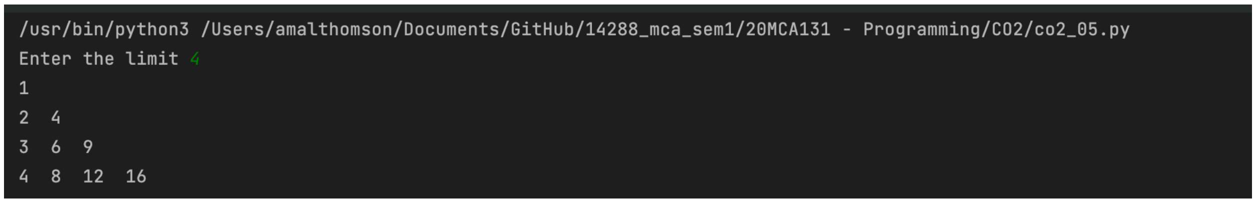
CO2

Implement decision making, looping constructs and functions

Procedure

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

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_05.py
Enter the limit 4
1
2  4
3  6  9
4  8  12  16
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 25

Aim

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

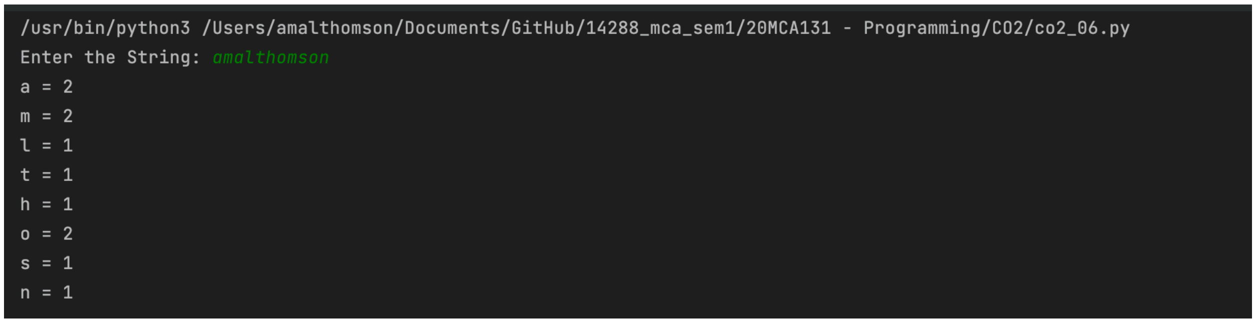
CO2

Implement decision making, looping constructs and functions

Procedure

```
n=str(input("Enter the String: "))
s= {}
for i in n:
    if i in s:
        s[i]= s[i]+1
    else:
        s[i]=1
for m,s in s.items():
    print(m ,"=",s)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_06.py
Enter the String: amalthomson
a = 2
m = 2
l = 1
t = 1
h = 1
o = 2
s = 1
n = 1
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 26

Aim

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

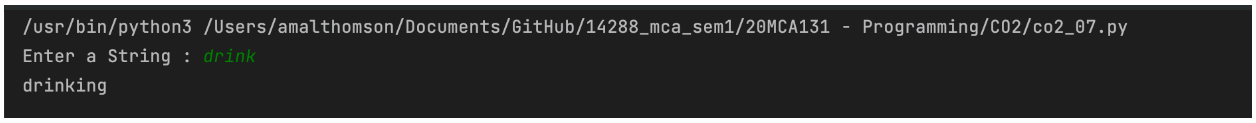
CO2

Implement decision making, looping constructs and functions

Procedure

```
x=str(input("Enter a String : "))
if(x[-3:]=="ing"):
   x=x+"ly"
else:
   x=x+"ing"
print(x)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_07.py
Enter a String : drink
drinking
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 27

Aim

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

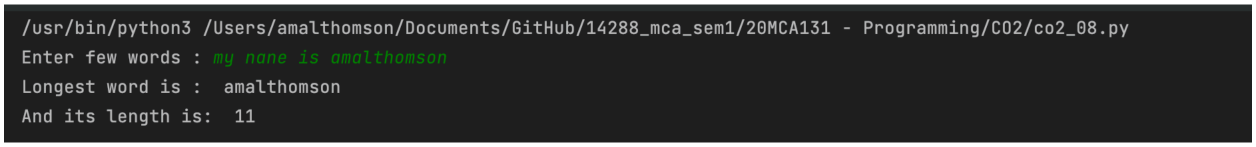
CO2

Implement decision making, looping constructs and functions

Procedure

```
sen = input("Enter few words : ")
longest = max(sen.split(), key=len)
print("Longest word is : ", longest)
print("And its length is: ", len(longest))
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_08.py
Enter few words : my nane is amalthomson
Longest word is :  amalthomson
And its length is:  11
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 28

Aim

Construct following pattern using nested loop

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

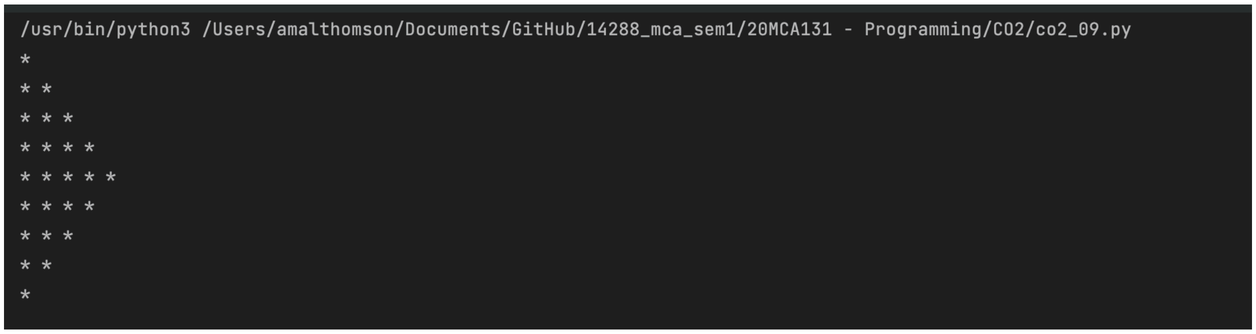
CO2

Implement decision making, looping constructs and functions

Procedure

```
for i in range(1,5):
    for j in range(i):
        print('* ', end="")
    print('')
for i in range(5, 0, -1):
    for j in range(i):
        print('* ', end="")
    print('')
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_09.py
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 29

Aim

Generate all factors of a number.

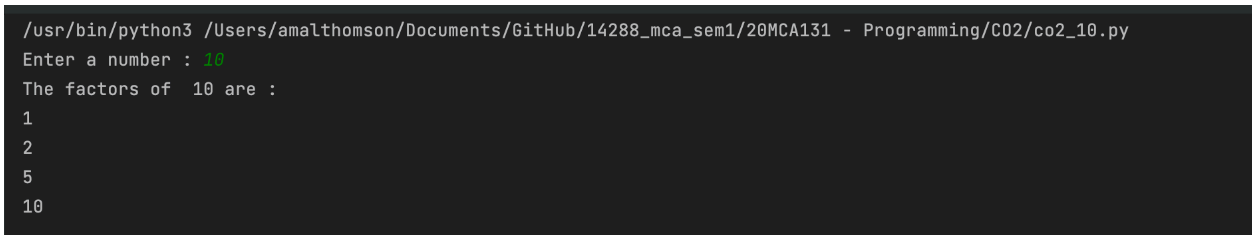
CO2

Implement decision making, looping constructs and functions

Procedure

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

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_10.py
Enter a number : 10
The factors of  10 are :
1
2
5
10
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 30

Aim

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

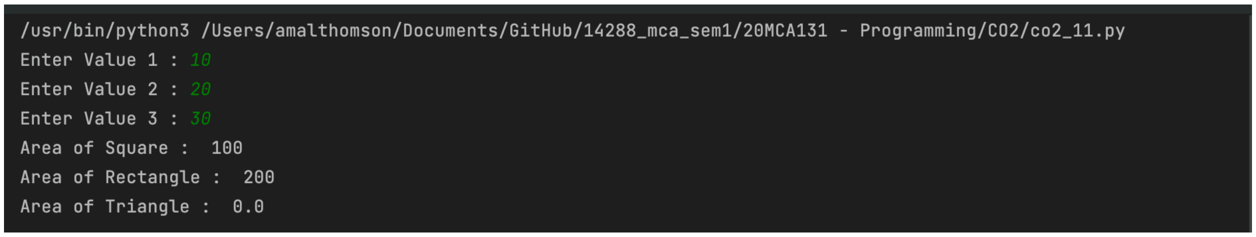
CO2

Implement decision making, looping constructs and functions

Procedure

```
a=int(input("Enter Value 1 : "))
b=int(input("Enter Value 2 : "))
c=int(input("Enter Value 3 : "))
area_square = lambda side : side * side
area_rectangle = lambda length,width : length * width
area_triangle =  lambda s,a,b,c : (s*(s-a)*(s-b)*(s-c)) ** 0.5
s = (a + b + c) / 2
print("Area of Square : ",area_square(a))
print("Area of Rectangle : ",area_rectangle(a,b))
print("Area of Triangle : ",area_triangle(s,a,b,c))
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO2/co2_11.py
Enter Value 1 : 10
Enter Value 2 : 20
Enter Value 3 : 30
Area of Square :  100
Area of Rectangle :  200
Area of Triangle :  0.0
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

Experiment No.: 31

Aim

Write a Python Program to subtract five days from the current date.

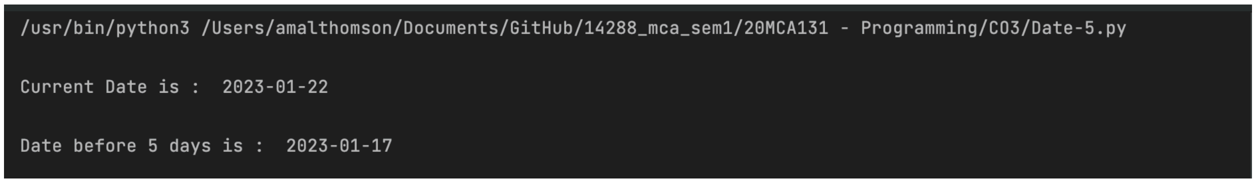
CO3

Design modules and packages - built in and user defined packages

Procedure

```
from datetime import date, timedelta
dt=date.today() - timedelta(5)
print("\nCurrent Date is : ", date.today())
print("\n Date before 5 days is : ", dt)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO3/Date-5.py

Current Date is :  2023-01-22

Date before 5 days is :  2023-01-17
```

Result

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 32

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.

CO3

Design modules and packages - built in and user defined packages

Procedure

circle.py

```
def area(radius):
    return radius * radius * 3.14
def perimeter(radius):
    return 2 * 3.14 * radius
```

rectangle.py

```
def area(length, breadth):
  return length * breadth
def perimeter(length, breadth):
  return 2 * (length + breadth)
```

sphere.py

```
def area(radius):
  return radius * radius * 3.14 * 4
def perimeter(radius):
  return 6.2832 * radius
```

cubiod.py

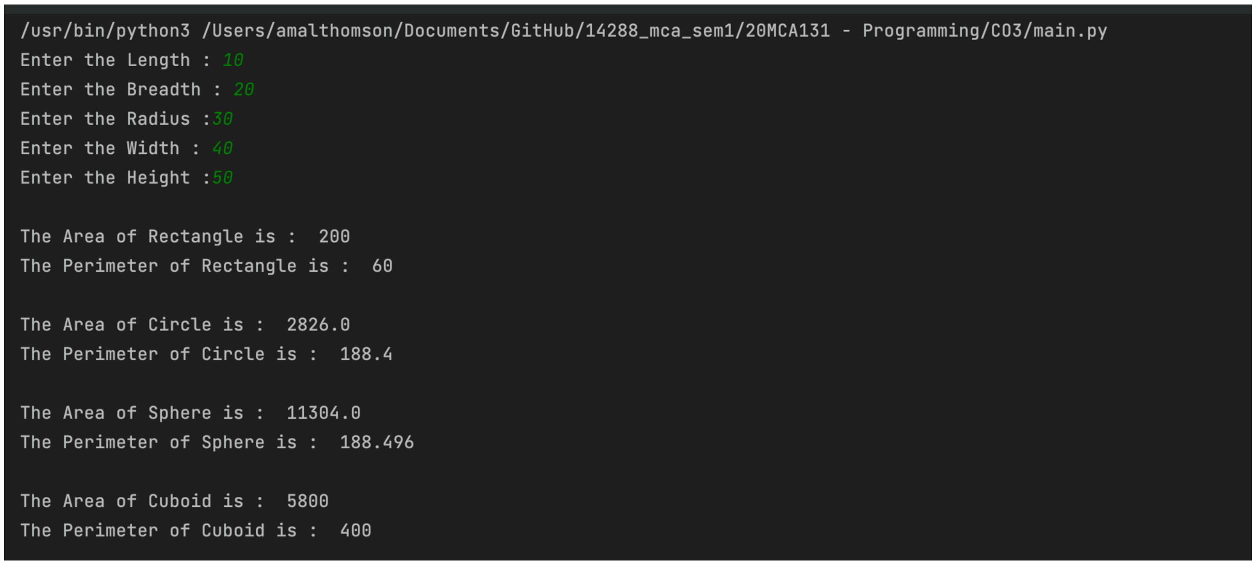
```
def area(length, width, height):
    return 2*(length*width + width*height + height*length)
def perimeter(length, width, height):
    return 4 * (length + width + height)
```

main.py

```
from graphics.rectangle import area, perimeter
length=int(input("Enter the Length : "))
breadth=int(input("Enter the Breadth : "))
radius=int(input("Enter the Radius :"))
width=int(input("Enter the Width : "))
height=int(input("Enter the Height :"))
arearect=area(length, breadth)
print("\nThe Area of Rectangle is : ", arearect)
perirect=perimeter(length, breadth)
print("The Perimeter of Rectangle is : ", perirect)
from graphics.circle import area, perimeter
areacir=area(radius)
print("\nThe Area of Circle is : ", areacir)
pericir=perimeter(radius)
print("The Perimeter of Circle is : ", pericir)
from graphics.graphics2.sphere import area, perimeter
areasph=area(radius)
print("\nThe Area of Sphere is : ", areasph)
perisph=perimeter(radius)
print("The Perimeter of Sphere is : ", perisph)
from graphics.graphics2.cubiod import area, perimeter
```

```
areacub=area(length, width, height)
print("\nThe Area of Cuboid is : ", areacub)
pericub=perimeter(length, width, height)
print("The Perimeter of Cuboid is : ", pericub)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO3/main.py
Enter the Length : 10
Enter the Breadth : 20
Enter the Radius :30
Enter the Width : 40
Enter the Height :50

The Area of Rectangle is :  200
The Perimeter of Rectangle is :  60

The Area of Circle is :  2826.0
The Perimeter of Circle is :  188.4

The Area of Sphere is :  11304.0
The Perimeter of Sphere is :  188.496

The Area of Cuboid is :  5800
The Perimeter of Cuboid is :  400
```

Result

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

Experiment No.: 33

Aim

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

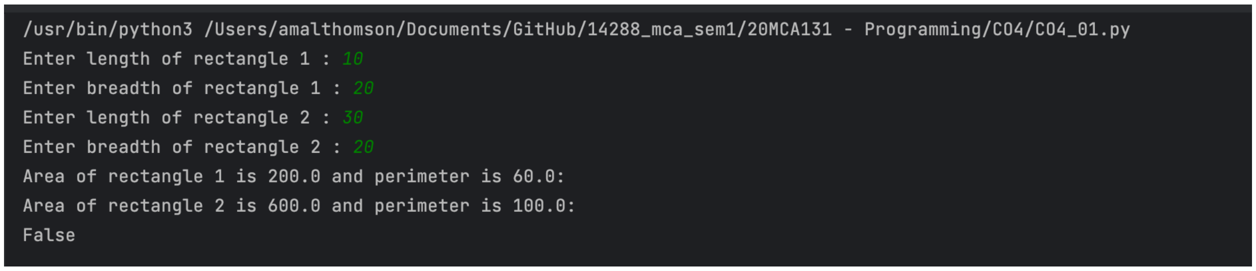
CO4

Implement object-oriented programming and exception handling.

Procedure

```
class Rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth
    def area(self):
        return self.length * self.breadth
    def perimeter(self):
        return 2 * (self.length + self.breadth)
l1 = float(input("Enter length of rectangle 1 : "))
b1 = float(input("Enter breadth of rectangle 1 : "))
l2 = float(input("Enter length of rectangle 2 : "))
b2 = float(input("Enter breadth of rectangle 2 : "))
rect1 = Rectangle(l1, b1)
rect2 = Rectangle(l2, b2)
print("Area of rectangle 1 is {} and perimeter is {}: ".format(rect1.area(), rect1.perimeter()))
print("Area of rectangle 2 is {} and perimeter is {}: ".format(rect2.area(), rect2.perimeter()))
print(rect1.area() > rect2.area())
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO4/CO4_01.py
Enter length of rectangle 1 : 10
Enter breadth of rectangle 1 : 20
Enter length of rectangle 2 : 30
Enter breadth of rectangle 2 : 20
Area of rectangle 1 is 200.0 and perimeter is 60.0:
Area of rectangle 2 is 600.0 and perimeter is 100.0:
False
```

Result

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 34

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.

CO4

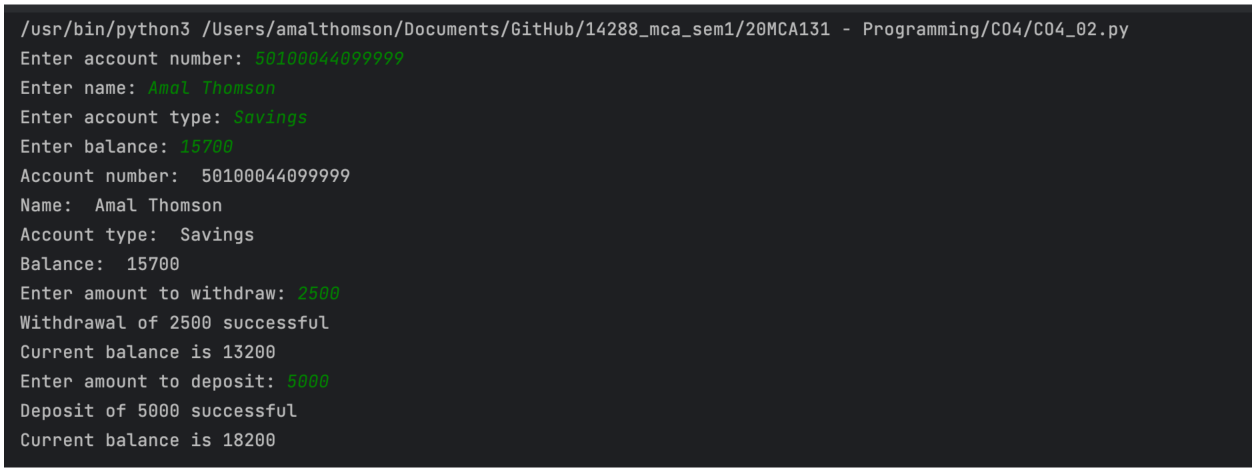
Implement object-oriented programming and exception handling.

Procedure

```
class Bank:
    def __init__(self, account_number, name, account_type, balance):
        self.account_number = account_number
        self.name = name
        self.account_type = account_type
        self.balance = balance
    def deposit(self, amount):
        self.balance += amount
        print("Deposit of {} successful".format(amount))
        print("Current balance is {}".format(self.balance))
    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance")
        else:
            self.balance -= amount
            print("Withdrawal of {} successful".format(amount))
            print("Current balance is {}".format(self.balance))
num = int(input("Enter account number: "))
```

```
name = input("Enter name: ")
acctype = input("Enter account type: ")
bal = int(input("Enter balance: "))
bnk = Bank(num, name, acctype, bal)
print("Account number: ", bnk.account_number)
print("Name: ", bnk.name)
print("Account type: ", bnk.account_type)
print("Balance: ", bnk.balance)
bnk.withdraw(int(input("Enter amount to withdraw: ")))
bnk.deposit(int(input("Enter amount to deposit: ")))
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO4/CO4_02.py
Enter account number: 50100044099999
Enter name: Amal Thomson
Enter account type: Savings
Enter balance: 15700
Account number:  50100044099999
Name:  Amal Thomson
Account type:  Savings
Balance:  15700
Enter amount to withdraw: 2500
Withdrawal of 2500 successful
Current balance is 13200
Enter amount to deposit: 5000
Deposit of 5000 successful
Current balance is 18200
```

Result

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 35

Aim

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

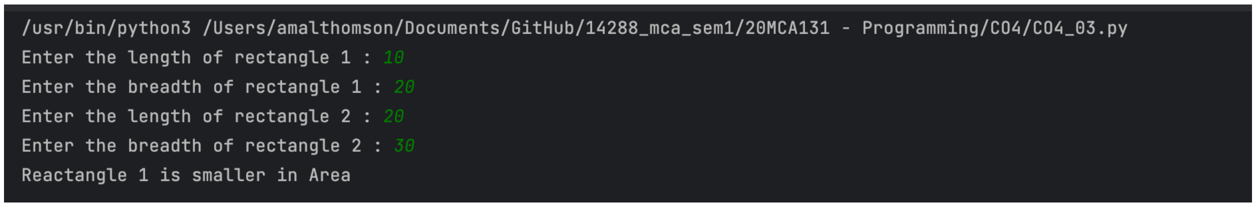
CO4

Implement object-oriented programming and exception handling.

Procedure

```
class Rectangle:
    def __init__(self, length, width):
        self.__length = length
        self.__width = width
        self.area=length*width
    def __lt__(self, other):
        if self.area<other.area:
            return "Reactangle 1 is smaller in Area"
        else:
            return "Reactangle 2 is smaller in Area"
l1=int(input("Enter the length of rectangle 1 : " ))
b1=int(input("Enter the breadth of rectangle 1 : " ))
l2=int(input("Enter the length of rectangle 2 : " ))
b2=int(input("Enter the breadth of rectangle 2 : " ))
r1=Rectangle(l1,b1)
r2=Rectangle(l2,b2)
print(r1<r2)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO4/CO4_03.py
Enter the length of rectangle 1 : 10
Enter the breadth of rectangle 1 : 20
Enter the length of rectangle 2 : 20
Enter the breadth of rectangle 2 : 30
Reactangle 1 is smaller in Area
```

Result

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 36

Aim

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

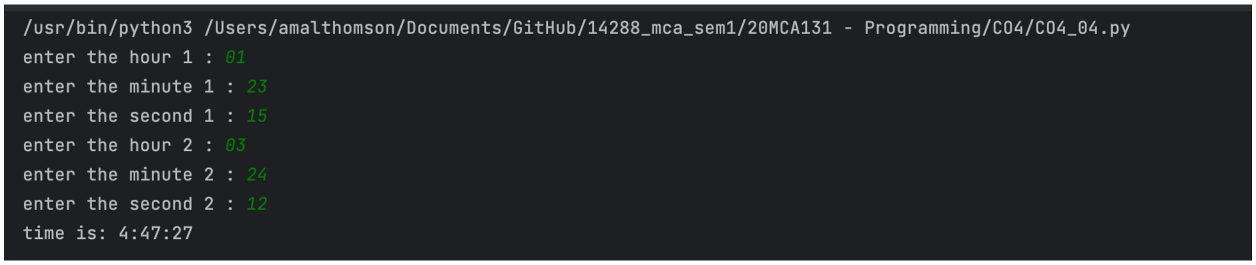
CO4

Implement object-oriented programming and exception handling.

Procedure

```
class Time:
    def __init__(self, hour, minute, second):
        self.__hour = hour
        self.__minute = minute
        self.__second = second
    def __add__(self, other):
        return 'time is: ' + str(self.__hour + other.__hour) + ':' + str(self.__minute + other.__minute)
+ ':' + str(
            self.__second + other.__second)
h = int(input("enter the hour 1 : "))
m = int(input("enter the minute 1 : "))
s = int(input("enter the second 1 : "))
h1 = int(input("enter the hour 2 : "))
m1 = int(input("enter the minute 2 : "))
s1 = int(input("enter the second 2 : "))
t1 = Time(h, m, s)
t2 = Time(h1, m1, s1)
print(t1 + t2)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO4/CO4_04.py
enter the hour 1 : 01
enter the minute 1 : 23
enter the second 1 : 15
enter the hour 2 : 03
enter the minute 2 : 24
enter the second 2 : 12
time is: 4:47:27
```

Result

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 37

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.

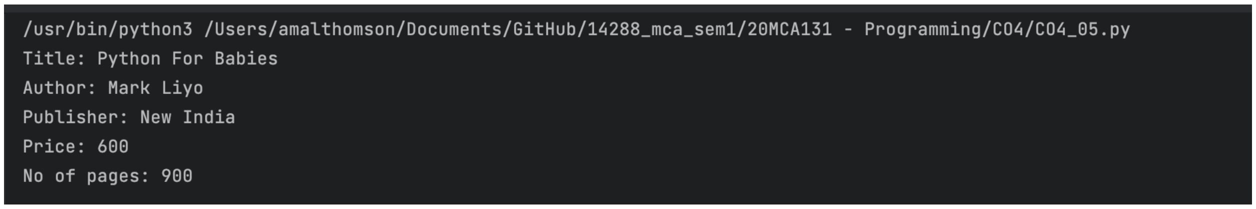
CO4

Implement object-oriented programming and exception handling.

Procedure

```
class Publisher:
    def __init__(self, Pubname):
        self.Pubname = Pubname
    def display(self):
        print("Publisher name is:", self.Pubname)
class Book(Publisher):
    def __init__(self, Pubname, title, author):
        Publisher.__init__(self, Pubname)
        self.title = title
        self.author = author
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
class Python(Book):
    def __init__(self, Pubname, title, author, price, no_of_pages):
        Book.__init__(self, Pubname, title, author)
        self.price = price
        self.no_of_pages = no_of_pages
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
        print("Publisher:",self.Pubname)
        print("Price:", self.price)
        print("No of pages:", self.no_of_pages)
b1 = Python("New India", "Python For Babies", "Mark Liyo", 600, 900)
b1.display()
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO4/CO4_05.py
Title: Python For Babies
Author: Mark Liyo
Publisher: New India
Price: 600
No of pages: 900
```

Result

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

Experiment No.: 38

Aim

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

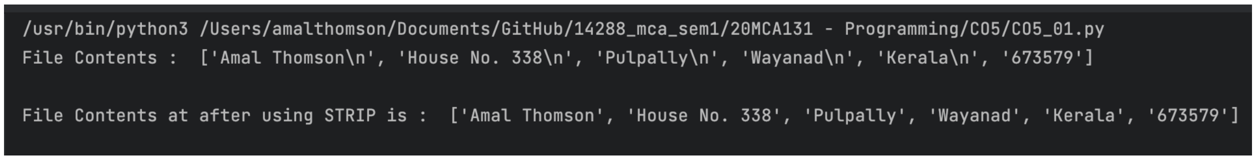
CO5

Create files and form regular expressions for effective search operations on strings and files.

Procedure

```
file = open ('sample.txt')
file2 = file.readlines()
print("File Contents : ", file2)
file3 = [X.strip() for X in file2]
print ("\nFile Contents at after using STRIP is : ", file3)
file.close()
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO5/CO5_01.py
File Contents :  ['Amal Thomson\n', 'House No. 338\n', 'Pulpally\n', 'Wayanad\n', 'Kerala\n', '673579']

File Contents at after using STRIP is :  ['Amal Thomson', 'House No. 338', 'Pulpally', 'Wayanad', 'Kerala', '673579']
```

Result

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 39

Aim

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

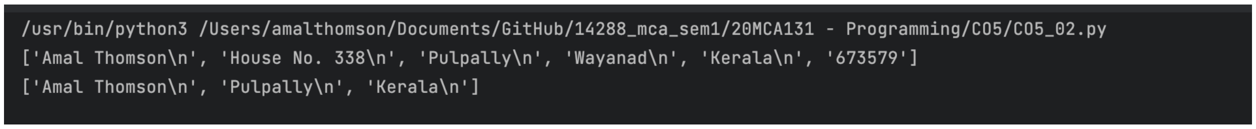
CO5

Create files and form regular expressions for effective search operations on strings and files.

Procedure

```
read_file = open("read.txt", "r")
read_lines = read_file.readlines()
print(read_lines)
read_file.close()
write_file = open("write.txt", "w")
for i in range(0, len(read_lines)):
    if i%2==0:
        write_file.write(read_lines[i])
write_file.close()
file = open("write.txt","r")
lines = file.readlines()
print(lines)
file.close()
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO5/CO5_02.py
['Amal Thomson\n', 'House No. 338\n', 'Pulpally\n', 'Wayanad\n', 'Kerala\n', '673579']
['Amal Thomson\n', 'Pulpally\n', 'Kerala\n']
```

Result

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 40

Aim

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

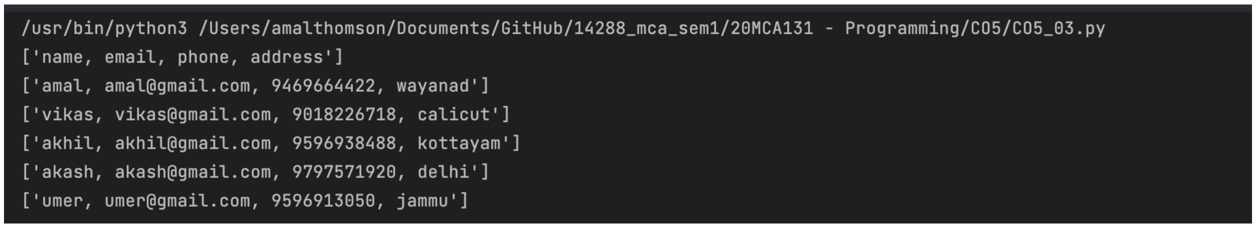
CO5

Create files and form regular expressions for effective search operations on strings and files.

Procedure

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

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO5/CO5_03.py
['name, email, phone, address']
['amal, amal@gmail.com, 9469664422, wayanad']
['vikas, vikas@gmail.com, 9018226718, calicut']
['akhil, akhil@gmail.com, 9596938488, kottayam']
['akash, akash@gmail.com, 9797571920, delhi']
['umer, umer@gmail.com, 9596913050, jammu']
```

Result

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 41

Aim

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

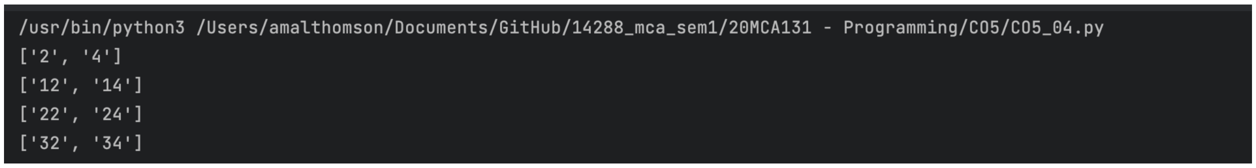
CO5

Create files and form regular expressions for effective search operations on strings and files.

Procedure

```
import csv
read_column = [1,3]
# opening csv file
with open('sample.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print([row[i] for i in read_column])
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO5/CO5_04.py
['2', '4']
['12', '14']
['22', '24']
['32', '34']
```

Result

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 42

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.

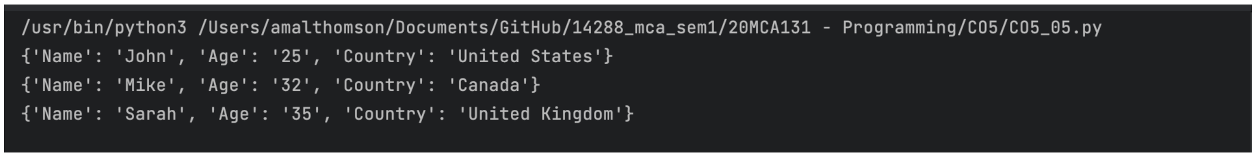
CO5

Create files and form regular expressions for effective search operations on strings and files.

Procedure

```
import csv
data = [{'Name': 'John', 'Age': 25, 'Country': 'United States'},
{'Name': 'Mike', 'Age': 32, 'Country': 'Canada'},
{'Name': 'Sarah', 'Age': 35, 'Country': 'United Kingdom'}]
with open('people.csv', 'w') as csvfile:
    headernames = ['Name', 'Age', 'Country']
    csvwriter = csv.DictWriter(csvfile, fieldnames=headernames)
    csvwriter.writeheader()
    for row in data:
        csvwriter.writerow(row)
with open('people.csv', 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
            print(row)
```

Output Screenshot



```
/usr/bin/python3 /Users/amalthomson/Documents/GitHub/14288_mca_sem1/20MCA131 - Programming/CO5/CO5_05.py
{'Name': 'John', 'Age': '25', 'Country': 'United States'}
{'Name': 'Mike', 'Age': '32', 'Country': 'Canada'}
{'Name': 'Sarah', 'Age': '35', 'Country': 'United Kingdom'}
```

Result

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

Experiment No.: 43

Aim

Micro Project – CRUD Operation using Django Framework (Employee Management).

CO

CO1,CO2,CO3,CO4,CO5

Procedure

manage.py

```
import os
import sys
def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'crud.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

forms.py

```
from django import forms
from employee.models import Employee
from django.forms import fields
class EmployeeForm(forms.ModelForm):
    class Meta:
        model = Employee
        fields = "__all__"
```

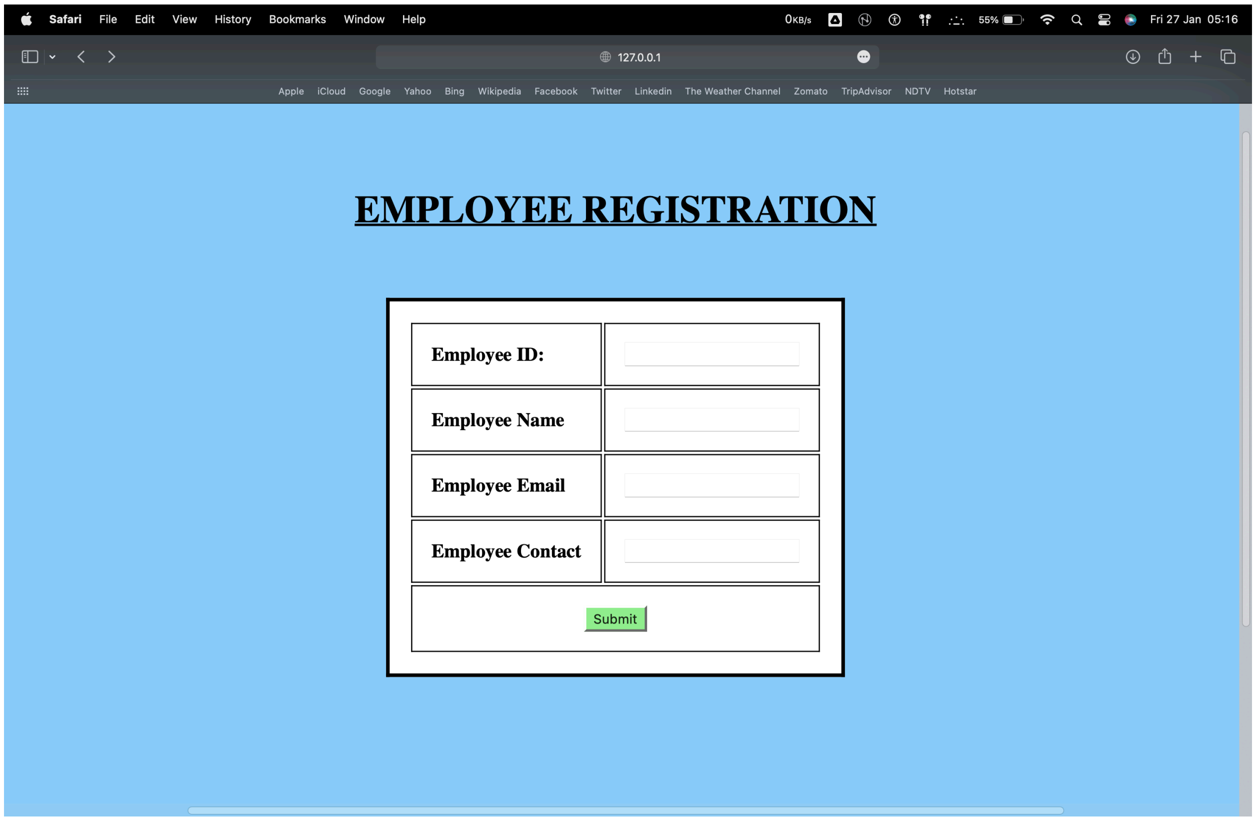
url.py

```
from django.contrib import admin
from django.urls import path
from employee import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('emp', views.emp),
    path('show',views.show),
    path('edit/<int:id>', views.edit),
    path('update/<int:id>', views.update),
    path('delete/<int:id>', views.destroy),
]
```

views.py

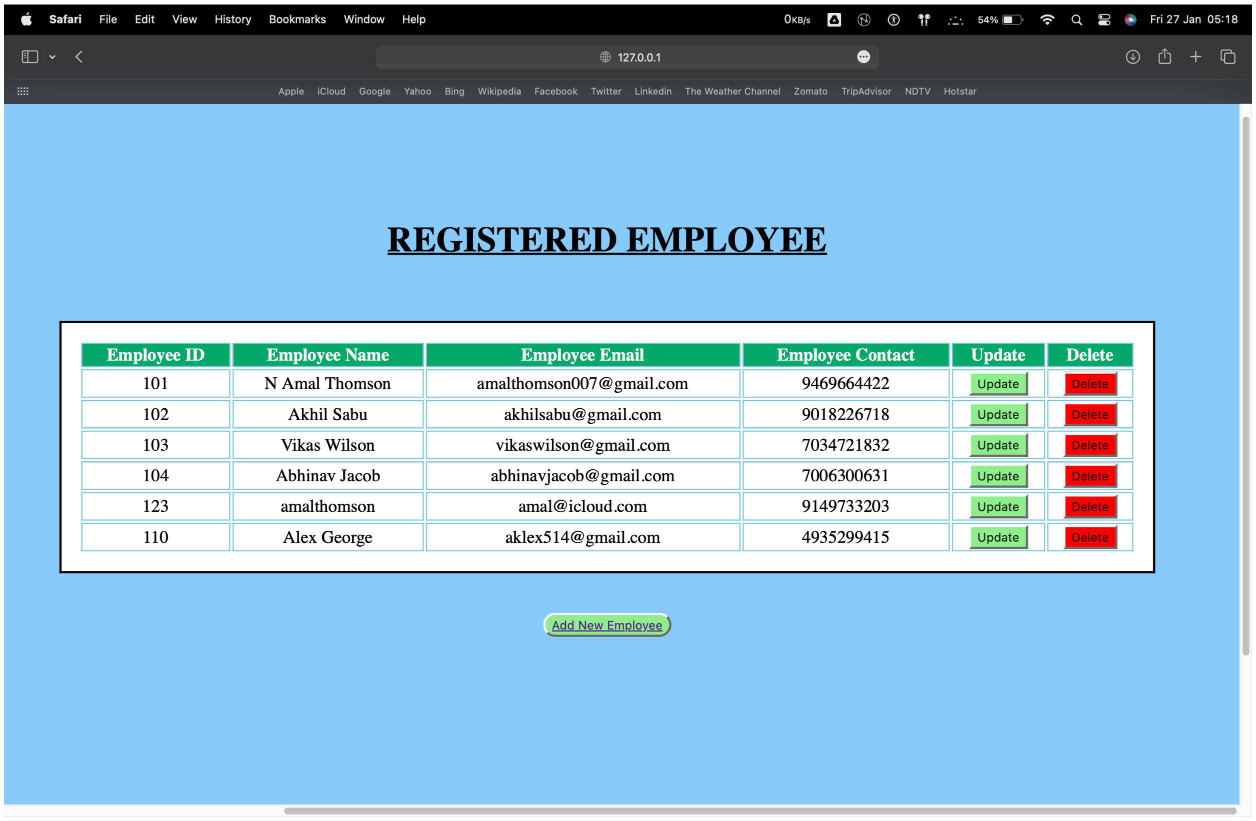
```
from django.shortcuts import render
from django.shortcuts import render, redirect
from employee.forms import EmployeeForm
from employee.models import Employee
def emp(request):
    if request.method == "POST":
        form = EmployeeForm(request.POST)
        if form.is_valid():
            try:
                form.save()
                return redirect('/show')
            except:
                pass
    else:
        form = EmployeeForm()
    return render(request,'index.html',{'form':form})
def show(request):
    employees = Employee.objects.all()
    return render(request,"show.html",{'employees':employees})
def edit(request, id):
    employee = Employee.objects.get(id=id)
    return render(request,'edit.html', {'employee':employee})
def update(request, id):
    employee = Employee.objects.get(id=id)
    form = EmployeeForm(request.POST, instance = employee)
    if form.is_valid():
        form.save()
        return redirect("/show")
    return render(request, 'edit.html', {'employee': employee})
def destroy(request, id):
    employee = Employee.objects.get(id=id)
    employee.delete()
    return redirect("/show")
```

Output Screenshot


Safari File Edit View History Bookmarks Window Help
Fri 27 Jan 05:16
127.0.0.1
Apple iCloud Google Yahoo Bing Wikipedia Facebook Twitter Linkedin The Weather Channel Zomato TripAdvisor NDTV Hotstar
Submit

EMPLOYEE REGISTRATION

Employee ID:	
Employee Name	
Employee Email	
Employee Contact	


Safari File Edit View History Bookmarks Window Help
Fri 27 Jan 05:18
127.0.0.1
Apple iCloud Google Yahoo Bing Wikipedia Facebook Twitter Linkedin The Weather Channel Zomato TripAdvisor NDTV Hotstar
Add New Employee

REGISTERED EMPLOYEE

Employee ID	Employee Name	Employee Email	Employee Contact	Update	Delete
101	N Amal Thomson	amalthomson007@gmail.com	9469664422	Update	Delete
102	Akhil Sabu	akhilsabu@gmail.com	9018226718	Update	Delete
103	Vikas Wilson	vikaswilson@gmail.com	7034721832	Update	Delete
104	Abhinav Jacob	abhinavjacob@gmail.com	7006300631	Update	Delete
123	amalthomson	amal@icloud.com	9149733203	Update	Delete
110	Alex George	aklex514@gmail.com	4935299415	Update	Delete

Result

The program was executed and the result was successfully obtained. Thus COs were obtained.