

MASTER OF COMPUTER APPLICATIONS

PRACTICAL RECORD WORK

ON

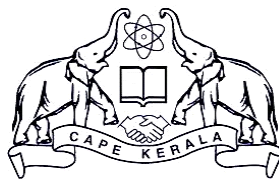
20MCA131 PROGRAMMING LAB

Submitted

By

JERIN S R

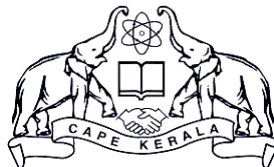
(Reg. No. : VDA20MCA-2036)



**DEPARTMENT OF COMPUTER APPLICATIONS
COLLEGE OF ENGINEERING VADAKARA
(CAPE - GOVT. OF KERALA)**

APRIL - 2021

**DEPARTMENT OF COMPUTER APPLICATIONS
COLLEGE OF ENGINEERING VADAKARA
(CAPE - GOVT. OF KERALA)**



CERTIFICATE

Certified that this is a bonafide record of the practical work on the course 20MCA131 PROGRAMMING LAB done by Mr. JERIN S R (Reg .No.:**VDA20MCA-2036**) First Semester MCA student of Department of Computer Applications at College of Engineering Vatakara in the partial fulfilment for the award of the degree of Master of Computer Applications (MCA) of APJ Abdul Kalam Technological University (KTU)

(Ms. Rajalekshmy K D)
FACULTY-IN-CHARGE

HEAD OF THE DEPARTMENT

CEV
22/04/2021

EXAMINERS:

INDEX			
SL.NO	PROGRAMS	PAGE NO	REMARKS
1	LEAP YEAR	3	
2	LIST COMPREHENSIONS	5	
3	OCCURRENCES OF WORD	8	
4	STORE “OVER”	10	
5	COUNT OF OCCURRENCE	12	
6	LIST OPERATIONS	14	
7	STRING REPLACEMENT	17	
8	CHARACTER EXCHANGE	19	
9	AREA OF CIRCLE	21	
10	BIGGEST OF 3 NUMBERS	23	
11	FILE EXTENTION	25	
12	LIST OF COLORS	27	
13	COMPUTE N+NN+NNN	29	
14	LIST INTERSECTION	31	
15	STRING SWAPPING	33	
16	DICTIONARY SORTING	35	
17	MERGE DICTIONARIES	37	
18	GCD	39	
19	REMOVING EVEN NUM	41	
20	FACTORIAL OF A NUM	43	
21	FIBONACCI	45	
22	SUM OF ALL TERMS LIST	47	
23	PERFECT SQUARE	49	
24	PYRAMID	51	
25	NUMBER OF CHAR IN STR	53	
26	ADDING “ING”, ”LY”	55	
27	LENGTH OF LONGEST WORD	57	

28	NESTED LOOP PATTERN	59	
29	FACTORS OF A NUMBER	61	
30	LAMBDA FUNCTIONS	63	
31	BUILT-IN PACKAGES	65	
32	USER DEFINED PACKAGE	67	
33	RECTANGLE WITH CLASS	72	
34	BANK ACCOUNT	74	
35	COMPARE AREA OF RECT	80	
36	TIME ADDING	83	
37	BOOK DETAILS	86	
38	READ FILE LINE BY LINE	89	
39	COPY ODD LINES -FILE	91	
40	READ ROW - CSV	93	
41	COLUMNS - CSV	95	
42	DICTIONARY TO CSV	97	

Experiment No 1

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

Source Code

leap_year.py

```
startYear = 2021
print("Enter any future year")
endYear = int(input())

print("List of leap years:")
for year in range(startYear, endYear):
    if (0 == year % 4) and (0 != year % 100) or (0 ==
year % 400):
        print(year)
```

OUTPUT

Enter any future year

2044

List of leap years:

2024

2028

2032

2036

2040

Experiment No 2

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

Source Code

list_operations.py

```
list1 = []
list2 = []
print("Select operation.")
print("1.Check Length of two list's are Equal")
print("2.Check sum of two list's are Equal")
print("3.whether any value occur in both ")
print("4.Display Lists")

while True:
    choice = input("Enter any choice ")
    if choice in ('1', '2', '3', '4'):
        list1Len = int(input("Enter the number of elements in list 1 : "))
        for i in range(0, list1Len):
            print("Enter the element ", i + 1, ":")
            item1 = int(input())
            list1.append(item1)
        list2Len = int(input("Enter the number of elements in list 2 : "))
        for j in range(0, list2Len):
            print("Enter the element ", j + 1, ":")
            item2 = int(input())
            list2.append(item2)

        if choice == '1':
            if len(list1) == len(list2):
                print(" Length are Equal")
            else:
                print(" Length are Not Equal")

        if choice == '2':
            if sum(list1) == sum(list2):
                print(" Sums are Equal")
            else:
                print(" Sums are Not Equal")
```

```
        if choice == '3':
            list3 = [x for x in list1 if x in list2]
            print("Common elements in both list's are
\n", list3)

        if choice == '4':
            print("List 1 is :\n", list1, " List 2 is :
\n", list2)
```


OUTPUT

Select operation.

- 1.Check Length of two list's are Equal
- 2.Check sum of two list's are Equal
- 3.whether any value occur in both
- 4.Display Lists

Enter any choice 1

Enter the number of elements in list 1 : 4

Enter the element 1 :

1

Enter the element 2 :

3

Enter the element 3 :

5

Enter the element 4 :

6

Enter the number of elements in list 2 : 4

Enter the element 1 :

3

Enter the element 2 :

4

Enter the element 3 :

6

Enter the element 4 :

7

Length are Equal

Enter any choice

Experiment No 3

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

Source Code

Occurences_Of_word_in_a_Text.py

```
text = input("\nEnter any text")
toFindWord = input("\nEnter a word to find")
a = []
count = 0
a = text.split(" ")
for i in range(0, len(a)):
    if toFindWord == a[i]:
        count = count + 1
print("\nCount of the word is:")
print(count)
```

OUTPUT

Enter any text hello friend is that lame

Enter a word to find enemy

Count of the word is:

0

Experiment No 4

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

Source Code

List_of_Integers.py

```
ListofInt = []
print(" This list can add up to 10 elements")
print("\n")
for i in range(1, 10):
    a = int(input("Enter the number "))
    if a < 100:
        ListofInt.append(a)
    else:
        ListofInt.append("over")
print(ListofInt)
```

OUTPUT

This list can add up to 10 elements

Enter the number 34

Enter the number 45

Enter the number 6

Enter the number 765

Enter the number 768

Enter the number 2

Enter the number 565

Enter the number 87

Enter the number 965

[34, 45, 6, 'over', 'over', 2, 'over', 87, 'over']

Experiment No 5

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

Source Code

List_of_firstnames_with_occurance.py

```
N = int(input("List length"))
listed = []
count = 0
for i in range(N):
    name = input("Enter name : ")
    listed.append(name)
for i in listed:
    for j in i:
        if j == "a":
            count = count+1
print(count)
```

OUTPUT

List length 4

Enter name : sansa

Enter name : arya

Enter name : jon

Enter name : rob

4

Experiment No 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

Source Code

list_operations_basic.py

```
list1 = []
list2 = []
print("Select operation.")
print("1.Check Length of two list's are Equal")
print("2.Check sum of two list's are Equal")
print("3.whether any value occur in both ")
print("4.Display Lists")

while True:
    choice = input("Enter any choice ")
    if choice in ('1', '2', '3', '4'):
        list1Len = int(input("Enter the number of elements in list 1 : "))
        for i in range(0, list1Len):
            print("Enter the element ", i + 1, ":")
            item1 = int(input())
            list1.append(item1)
        list2Len = int(input("Enter the number of elements in list 2 : "))
        for j in range(0, list2Len):
            print("Enter the element ", j + 1, ":")
            item2 = int(input())
            list2.append(item2)

        if choice == '1':
            if len(list1) == len(list2):
                print(" Length are Equal")
            else:
                print(" Length are Not Equal")

        if choice == '2':
            if sum(list1) == sum(list2):
                print(" Sums are Equal")
            else:
                print(" Sums are Not Equal")

        if choice == '3':
            list3 = [x for x in list1 if x in list2]
            print("Common elements in both list's are\n", list3)
```



```
        if choice == '4':
            print("List 1 is :\n", list1, " List 2 is :
\n", list2)
```

OUTPUT

Select operation.

- 1.Check Length of two list's are Equal
- 2.Check sum of two list's are Equal
- 3.whether any value occur in both
- 4.Display Lists

Enter any choice 1

Enter the number of elements in list 1 : 4

Enter the element 1 :

4

Enter the element 2 :

5

Enter the element 3 :

6

Enter the element 4 :

7

Enter the number of elements in list 2 : 5

Enter the element 1 :

3

Enter the element 2 :

6

Enter the element 3 :

7

Enter the element 4 :

5

Enter the element 5 :

7

Length are Not Equal

Enter any choice

Experiment No 7

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

[eg: onion -> oni\$n]

Source Code

Replace_repeating_letter_with_\$.py

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

OUTPUT

Enter the word run baby run

run\$baby\$run

Experiment No 8

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

Source Code

```
string = input("Enter a string : ")
print("\n String after replacing first and last character", string[-1] + string[1:-1] + string[0])
```

OUTPUT

Enter a string : justin

String after replacing first and last character nustij

Experiment No 9

Accept the radius from user and find area of circle.

Source Code

Area_of_Circle.py

```
# normal way

PI = 3.14

r = float(input("Enter the radius of the circle: "))

area = PI * r * r

print("%.2f" % area)


# best way

import math as ma

area = ma.pi * pow(r, 2)

print("area is", area)

# use any of it
```

OUTPUT

Enter the radius of the circle: 8

area is 201.06192982974676

Experiment No 10

Find biggest of 3 numbers entered.

Source Code

Biggest_of_three.py

```
num1 = float(input("\nEnter the first number\n"))
num2 = float(input("\nEnter the second number\n"))
num3 = float(input("\nEnter the third number\n"))

if (num1 > num2) and (num1 > num3):
    max = num1

elif (num2 > num1) and (num2 > num3):
    max = num2

else:
    max = num3

print("the largest among three numbers is", max)
```

OUTPUT

Enter the first number

5

Enter the second number

3

Enter the third number

7

the largest among three numbers is 7.0

Experiment No 11

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

Source Code

File_extention.py

```
filename = input('Enter a filename: ')
index = 0
for i in range(len(filename)):
    if filename[i] == '.':
        index = i
print("File extension is ", filename[index + 1:])
```

OUTPUT

Enter a filename: sample.pdf

File extension is pdf

Experiment No 12

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

Source Code

List_of_colors.py

```
color = input("Enter the list of color names separated  
by commas:")  
  
lst1 = color.split(",")  
  
print("First color entered :", lst1[0])  
  
print("Last color entered :", lst1[-1])
```

OUTPUT

Enter the list of color names separated by commas: blue
,red, black

First color entered : blue

Last color entered : black

Experiment No 13

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

Source Code

Accept_and_computing_with_n.py

```
color = input("Enter the list of color names separated  
by commas:")  
  
lst1 = color.split(",")  
  
print("First color entered :", lst1[0])  
  
print("Last color entered :", lst1[-1])
```

OUTPUT

Enter the number 646

The value is: 647293938

Experiment No 14

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

Source Code

Colors_list_inbuilt.py

```
colorList1 = []

colorList2 = []

colorList1Count = int(input("Total elements in list one :"))

for i in range(colorList1Count):

    value = input("Enter a color")

    colorList1.append(value)

colorList2Count = int(input("Total elements in list two :"))

for i in range(colorList2Count):

    value = input("Enter a color : ")

    colorList2.append(value)

set1=set(colorList1)

set2 =set(colorList2)

print(set1.difference(set2))
```

OUTPUT

Total elements in list one :2

Enter a color blue

Enter a color black

Total elements in list two : 3

Enter a color : red

Enter a color : green

Enter a color : blue

{'blue ', ' black '}

Experiment No 15

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

Source Code

```
string = input("Enter 2 string separated by space : ")
string = string.split(' ')

# swapping two at position 1

print(string[0][0] + string[1][1] + string[0][2:] + " "
      + string[1][0] + string[0][1] + string[1][2:])

# swapping and making the swapped item UPPERCASE

print(string[0][0] + string[1][1].upper() + string[0][2:]
      + " " + string[1][0] + string[0][1].upper() +
      string[1][2:])
```

OUTPUT

Enter 2 string separated by space : kim wexler

kem wixler

kEm wIxler

Experiment No 16

Sort dictionary in ascending and descending order.

Source Code

Sort_dictionary.py

```
dict1 = {}

print(type(dict1))

limit = int(input("Enter the limit"))

for i in range(limit) :

    dict1.update({input("Enter the key") : input("Enter
the value")})

# dictItem = {'apple': 40, 'orange': 2, 'banana': 1,
'lemon': 3}

l = list(dict1.items()) # dict to list conversion

l.sort()

print("\n Ascending order is", l) # sorted list

l = list(dict1.items())

l.sort(reverse=True) # sorting in reverse order

print("\nDescending order is", l)

dict = dict(l) # list to dict

print("\nDictionary", dict)

print(("dict is" , dict1))
```

OUTPUT

Enter the limit 3

Enter the key 1

Enter the value one

Enter the key 2

Enter the value two

Enter the key 3

Enter the value three

Ascending order is [(' 1', ' one'), (' 2', ' two'), (' 3', ' three')]

Descending order is [(' 3', ' three'), (' 2', ' two'), (' 1', ' one')]

Dictionary {' 3': ' three', ' 2': ' two', ' 1': ' one'}
('dict is', {' 1': ' one', ' 2': ' two', ' 3': ' three'})

Experiment No 17

Merge two dictionaries

Source Code

Merge_dictionary.py

```
def Merge(dict1, dict2):  
    return dict2.update(dict1)
```

```
dict1 = {'a': 10, 'b': 8}
```

```
dict2 = {'d': 6, 'c': 4}
```

```
Merge(dict1, dict2)
```

```
print("Merge done \n ", dict2)
```

OUTPUT

Merge done

```
{'d': 6, 'c': 4, 'a': 10, 'b': 8}
```


Experiment No 18

Find GCD of two numbers

Source Code

GCD.py

```
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 += 1
print("Greatest common divisor is", gcd)
```

OUTPUT

Enter 1st number: 5

Enter 2nd number: 7

Greatest common divisor is 1

Experiment No 19

From a list of integers create a list removing even numbers

Source Code

Removing_Even_numbers_in_LIST.py

```
numberList = []

even_numberList = []

n = int(input("Enter the number of elements  "))


print("\n")

for i in range(0, n):

    print("Enter the element ", i + 1, ":")

    item = int(input())

    numberList.append(item)

print(" List is ", numberList)

even_numberList = [x for x in numberList if x % 2 != 0]

print("\n")

print("List after removing even numbers\n ", even_numberList)
```

OUTPUT

Enter the number of elements 5

Enter the element 1 :

4

Enter the element 2 :

6

Enter the element 3 :

7

Enter the element 4 :

8

Enter the element 5 :

5

List is [4, 6, 7, 8, 5]

List after removing even numbers

[7, 5]

Experiment No 20

Program to find factorial of a number

Source Code

Factorial.py

```
num = int(input("enter a number: "))

fac = 1

i = 1

while i <= num:
    fac = fac * i
    i = i + 1

print("factorial of ", num, " is ", fac)
```

OUTPUT

enter a number: 6

factorial of 6 is 720

Experiment No 21

Generate Fibonacci series of n terms

Source Code

Fibonacci.py

```
def fibonacci(n):  
    a, b = 0, 1  
    while a < n:  
        print(a)  
        a, b = b, a + b  
  
n = int(input("Enter the LIMIT"))  
  
fibonacci(n)
```

OUTPUT

Enter the LIMIT 5

0

1

1

2

3

Experiment No 22

Sum of all terms in a list

Source Code

Sum_of_list.py

```
n = int(input("Enter the N th Digit "))  
  
sum_num = (n * (n + 1)) / 2  
  
print(sum_num)
```

OUTPUT

Enter the N th Digit 5

15.0

Experiment No 23

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

Source Code

```
import math

start = int(input("Enter a starting range in 4 digit:"))

end = int(input("Enter an ending range in 4 digit:"))

perfect = []

for i in range(start, end + 1):

    flag = 0

    num = i

    while num > 0:

        digit = num % 10

        if digit not in [0, 2, 4, 6, 8]:

            flag = 1

            break

        num = int(num / 10)

    if flag == 0 and math.sqrt(i) % 1 == 0:

        perfect.append(i)

print("The list of perfect square numbers are:", perfect)
```

OUTPUT

Enter a starting range in 4 digit: 4969

Enter an ending range in 4 digit:9504

The list of perfect square numbers are: [6084, 6400, 8464]

Experiment No 24

Display The Given Pyramid With Step Number Accepted From User

Eg: N=4

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

Source Code

Pattern.Py

```
Def Pattern(Number):
    For I In Range(1, Number + 1):
        For J In Range(1, I + 1):
            Print(J * I, End=" ")
        Print("")
```

```
N = Int(Input("Limit : "))
Pattern(N)
```

OUTPUT

Limit : 6

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

Experiment No 25

Count The Number Of Characters (Character Frequency) In A String

Source Code

Number_Of_Characters.Py

```
Integer = Int(Input('Enter An Integer :'))  
String = Str(Integer)  
Print(Len(String))
```

OUTPUT

Enter an integer : 3743

4

Experiment No 26

Add 'ing' at the end of given string if it already exist with 'ing' then add 'ly'

Source Code

String_Ends_With.Py

```
String = Input("Enter The Text")

If Len(String) < 3:
    Print(String)
Elif String.EndsWith('Ing'):
    Print(String + 'Ly')
Else:
    Print(String + 'Ing')
```

OUTPUT

Enter the TEXT running
runningly

Experiment No 27

Accept the list of words and return the length of the longest word

Source Code

Length_Of_The_Longest_Word.Py

```
Def Longest(Get_Name):  
    Length = 0  
    Get_Name = Get_Name.Split()  
    For I In Get_Name:  
        If Len(I) > Length:  
            Length = Len(I)  
    Return Length  
  
Str = Input("Enter The String: ")  
Word = Longest(Str)  
Print(Word)
```

OUTPUT

Enter the string: Schemes and plots are the same thing

7

Experiment No 28

Construct following pattern using nested loop

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

Source Code

Pattern.Py

```
N = 5;
For I In Range(N):
    For J In Range(I):
        Print('* ', End="")
    Print('\n')

For I In Range(N, 0, -1):
    For J In Range(I):
        Print('* ', End="")
    Print('\n')
```

OUTPUT

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

Experiment No 29

Generate all factors of a number

Source Code

Factors_Of_Number.Py

```
Num = Int(Input("Enter A Number"))
Factors = []
For I In Range(1, Num + 1):
    If Num % I == 0:
        Factors.Append(I)

Print("Factors Of {} = {}".Format(Num, Factors))
```

OUTPUT

Enter a number 5

Factors of 5 = [1, 5]

Experiment No 30

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

Source Code

Lambda_Function.Py

```
L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
Print(List(Filter(Lambda N: N % 5 == 0, L)))

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

A = 10
B = 20
C = 15
S = (A + B + C) / 2

Print(Area_Square(A))
Print(Area_Rectangle(A, B))
Print(Area_Triangle(S, A, B, C))
```

OUTPUT

[5, 10, 15]

100

200

72.61843774138907

Experiment No 31

Work with built-in packages

Source Code

Builtfunc.Py

```
Import Math
    Num = Int(Input("Enter A Number: "))
    P = Int(Input("Enter The Power: "))
    Print(Num," Power ", P, " Is: ", Math.Pow(Num, P))
    Print("Square Root Of 64:",Math.Sqrt(64))
```

OUTPUT

Enter a number: 4

Enter the power: 6

4 power 6 is: 4096.0

square root of 64: 8.0

Experiment No 32

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)

Source Code

Main.Py

```
From Graphics Import Circle As C
From Graphics Import Rectangle As R
From Graphics.Threedg Import Cuboid As Cd
From Graphics.Threedg Import Sphere As Sp

# User Input
While True:
    Print("\n♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ Select Any To Perform
Operation ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦\n")
    Option = Int(Input("\N 1. Rectangle \N 2.Circle \N
3.Cuboid \N 4.Sphere\n"))
    Print("♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ Select Any To Perform
Operation ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦")
    If Option == 1:
        Print("♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ Rectangle Operations ♦ ♦
♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦")
        Rectoption = Int(Input("\N 1.Area \N
2.Perimeter \N"))
        If Rectoption == 1:
            Length = Int(Input("Enter The Length Of
Rectangle \T"))
            Breadth = Int(Input("Enter The Breadth
Rectangle\T"))
            Print("Area Is ", R.Area(Length, Breadth))
        Elif Rectoption == 2:
            Length = Int(Input("Enter The Length Of
Rectangle \T"))
            Breadth = Int(Input("Enter The Breadth Of
Rectangle\T"))
            Print("Perimeter Is ", R.Perimeter(Length,
Breadth))
        Else:
            Print("Invalid Option !")
    If Option == 2:
```

```

        Print("❖ ❖ ❖ ❖ ❖ ❖ Circle Operations ❖ ❖ ❖ ❖
❖ ❖ ❖ ❖ ❖ ❖")
        Circoption = Int(Input("\N 1.Area \N
2.Perimeter \N"))
        If Circoption == 1:
            R = Int(Input("Enter The Radius \T"))
            Print("Area Of Circle With Radius ", R, "Is
", C.Area(R))
        Elif Circoption == 2:
            R = Int(Input("Enter The Radius \T"))
            Print("Perimeter Of The Circle With Radius
", R, "Is", C.Perimeter(R))
        Else:
            Print("Invalid Option !")
    If Option == 3:
        Print("❖ ❖ ❖ ❖ ❖ ❖ Cuboid Operations ❖ ❖ ❖ ❖
❖ ❖ ❖ ❖ ❖ ❖")
        Quboption = Int(Input("\N 1.Area \N 2.Perimeter
\N 3.Volume \N"))
        If Quboption == 1:
            Length = Int(Input("Enter The Length Of
Cuboid \T"))
            Width = Int(Input("Enter The Height Of
Cuboid\T"))
            Height = Int(Input("Enter The Breadth Of
Cuboid\T"))
            Print("Surface Area Of Cuboid Is ",
Cd.Area(Length, Width, Height))
        Elif Quboption == 2:
            Length = Int(Input("Enter The Length Of
Cuboid\T"))
            Width = Int(Input("Enter The Height Of
Cuboid\T"))
            Height = Int(Input("Enter The Breadth Of
Cuboid\T"))
            Print("Perimeter Of Cuboid Is ",
Cd.Perimeter(Length, Width, Height))
        Elif Quboption == 3:
            Length = Int(Input("Enter The Length Of
Cuboid\T"))
            Width = Int(Input("Enter The Height Of
Cuboid\T"))
            Height = Int(Input("Enter The Breadth Of
Cuboid\T"))
            Print("Volume Of Cuboid Is ",
Cd.Volume(Length, Width, Height))
        Else:
            Print("Invalid Option !")

```

```

        If Option == 3:
            Print("❖ ❖ ❖ ❖ ❖ ❖ ❖ Sphere Operations ❖ ❖ ❖ ❖
❖ ❖ ❖ ❖ ❖ ❖")
            Sphoption = Int(Input("\N 1.Area \N 2.Perimeter
\N 3.Volume \N"))
            If Sphoption == 1:
                R = Int(Input("Enter The Radius\T"))
                Print("Area Of Sphere With Radius ", R,
"Is", Sp.Area(R))
            Elif Sphoption == 2:
                R = Int(Input("Enter The Radius\T"))
                Print("Perimeter Of Sphere With Radius ",
R, "Is", Sp.Perimeter(R))
            Elif Sphoption == 3:
                R = Int(Input("Enter The Radius\T"))
                Print("Volume Of Sphere With Radius ", R,
"Is", Sp.Volume(R))
            Else:
                Print("Invalid Option !")

```

Graphics

Circle.Py

```

Import Math

```

```

Def Area(R):
    Return Math.Pi * Math.Pow(R, 2)

```

```

Def Perimeter(R):
    Return 2 * Math.Pi * R

```

Rectangle.Py

```

Def Area(Length, Breadth):
    Return Length * Breadth

```

```

Def Perimeter(Length, Breadth):
    Return 2 * (Length + Breadth)

```

Threedg

Cuboid.Py

```
Def Area(Length, Width, Height):
    Length = Int(Input("Enter The Length Of Cuboid"))
    Height = Int(Input("Enter The Breadth Of Cuboid"))
    Width = Int(Input("Enter The Height Of Cuboid"))
    Return 2 * (Length * Width + Width * Height +
Height * Length)
```

```
Def Perimeter(Length, Width, Height):
    Length = Int(Input("Enter The Length Of Cuboid"))
    Height = Int(Input("Enter The Breadth Of Cuboid"))
    Width = Int(Input("Enter The Height Of Cuboid "))
    Return 4 * (Length + Width + Height)
```

```
Def Volume(Length, Width, Height):
    Return Length * Width * Height
```

Sphere.Py

```
Import Math As M
```

```
Def Area(R):
    Return 4 * M.Pi * M.Pow(R, 2)
```

```
Def Perimeter(R):
    Return 4 * M.Pi * M.Pow(R, 3)
```

```
Def Volume(R):
    Return 4.0 / 3.0 * M.Pi * M.Pow(R, 3)
```


OUTPUT

◆ ◆ ◆ ◆ ◆ ◆ ◆ SELECT ANY TO PERFORM OPERATION ◆ ◆ ◆ ◆

1. Rectangle
- 2.Circle
- 3.Cuboid
- 4.Sphere

1

◆ ◆ ◆ ◆ ◆ ◆ ◆ SELECT ANY TO PERFORM OPERATION ◆ ◆ ◆ ◆

◆ ◆ ◆ ◆ ◆ ◆ ◆ RECTANGLE OPERATIONS ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

1.Area

2.Perimeter

1

Enter the length of Rectangle 3

Enter the Breadth Rectangle 4

Area is 12

Invalid Option !

◆ ◆ ◆ ◆ ◆ ◆ ◆ SELECT ANY TO PERFORM OPERATION ◆ ◆ ◆ ◆

1. Rectangle
- 2.Circle
- 3.Cuboid
- 4.Sphere

Experiment No 33

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

Source Code

Rectangle_Class.Py

```
Class Rectangle:
    __Width = 0
    __Length = 0
    __Area = 0
    __Perimeter = 0

    Def __Init__(Self, Length, Width):
        Self.__Length = Length
        Self.__Width = Width

    Def Calc_Area(Self):
        Self.__Area = Self.__Length * Self.__Width
        Print("Area Is  :", Self.__Area)

    Def Calc_Perimeter(Self):
        Self.__Perimeter = 2 * (Self.__Length +
Self.__Width)
        Print("Perimeter Is :", Self.__Perimeter)

While True:
    Length = Int(Input("Enter Length Of The Rectangle :
"))
    Width = Int(Input("Enter Width Of The Rectangle :
"))
    Obj = Rectangle(Length, Width)
    Opt = Input(" [ A ]To Find Area  \N[ P ]To Find
Perimeter \Noption  :: ")
    If Opt == 'A' Or 'A':
        Obj.Calc_Area()
    Elif Opt == 'P' Or 'P':
        Obj.Calc_Perimeter()
    Else:
        Print("Options Are Wrong !!")
```

OUTPUT

Enter length of the rectangle : 3

Enter width of the rectangle : 5

[A]To find Area

[P]To find Perimeter

option :: a

Area is : 15

Enter length of the rectangle :

Experiment No 34

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

Source Code

Bank_account.py

```
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 deposit(self, a_deposit):

        print("Initial balance is  :", self.__acc_bal-
ance)

        print("Deposit is  :", a_deposit)

        self.__acc_balance += a_deposit

        print("Current balance is  :", self.__acc_bal-
ance)

    def withdraw(self):

        print("Current balance is  :", self.__acc_bal-
ance)
```

```

        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.__ac-
c_balance)

        else:

            print(self.amount, " is withdrawn .")

            self.__acc_balance -= self.amount

            print("Current balance is  : ", self.__ac-
c_balance)


    def acc_info(self):

        print(

"\n\n|||||
|||||
|||||\n\n")

        print("Account holder name  : ", self.__acc_-
name)

        print("Account number        : ", self.__ac-
c_no)

        print("Account type          : ",
self.__acc_type)

        print("Account Balance is    : ", self.__ac-
c_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)Deposit \n2)Withdraw \n3)Ac-
count info \n0)Exit\nChoose your option :: "))

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

        if opt == 1:

            amount = int(input("Deposit amount : "))

            holder.deposit(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

Enter Account holder name : jerry
Enter Account number : 74743040
Enter Account type : savings
Enter Account initial balance : 4747

.....
..

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

.....
..

Deposit amount : 333
Initial balance is : 4747
Deposit is : 333
Current balance is : 5080

.....
..

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

.....
..

Current balance is : 5080
How much amount need to withdraw : 44
44 is withrawed .
Current balance is : 5036

.....
..

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

Experiment No 35

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

Source Code

Rectangle private.py

```
# purpose - Create a class rectangle with private attributes length and width. overload '<' operator to compare the
```

```
# area of two reactangles
```

```
class Rectangle:
```

```
    def __init__(self, length, width):
```

```
        self.__length = length
```

```
        self.__width = width
```

```
    def area(self):
```

```
        return self.__length * self.__width
```

```
    def __lt__(self, other):
```

```
        return self.area() < other.area()
```

```
l1 = int(input("Enter the length of first rectangle"))
```

```
w1 = int(input("Enter the width of second rectangle"))
```

```
rectangle1 = (l1, w1)
```

```
l2 = int(input("Enter the length of first rectangle"))
```

```
w2 = int(input("Enter the width of second rectangle"))
```

```
rectangle2 = (l2, w2)
```

```
# r1 = Rectangle(4, 5)
```

```
# r2 = Rectangle(6, 3)

if rectangle1 < rectangle2:
    print("Area of rectangle 1 is smaller")
else:
    print("Area of rectangle 2 is smaller")
```

OUTPUT

Enter the length of first rectangle 5

Enter the width of second rectangle 7

Enter the length of first rectangle 7

Enter the width of second rectangle 8

Area of rectangle 1 is smaller

Experiment No 36

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

Source Code

Time_adding.py

```
class Time:

    def __init__(self, hour, minute, second):

        self.__hour = hour

        self.__minute = minute

        self.__second = second

    def __add__(self, other):

        second = self.__second + other.__second

        minute = self.__minute + other.__minute

        hour = self.__hour + other.__hour

        if second > 60:

            minute += int(second / 60)

            second = second % 60

        if minute > 60:

            hour += int(minute / 60)

            minute = minute % 60

        time = "{0} Hours: {1} minutes: {2} seconds".format(hour, minute, second)

        return time

h1 = int(input("Enter the Hours of first TIME \t"))

m1 = int(input("Enter the Minutes of first TIME \t"))
```

```
s1 = int(input("Enter the Seconds of first TIME \t"))

h2 = int(input("Enter the Hours of second TIME\t"))
m2 = int(input("Enter the Minutes of second TIME\t"))
s2 = int(input("Enter the Seconds of second TIME\t"))

time1 = Time(h1, m1, s1)
time2 = Time(h2, m2, s2)

print("Sum of time:", time1+time2)
```

OUTPUT

Enter the Hours of first TIME	4
Enter the Minutes of first TIME	6
Enter the Seconds of first TIME	3
Enter the Hours of second TIME	7
Enter the Minutes of second TIME	8
Enter the Seconds of second TIME	4
Sum of time: 11 Hours: 14 minutes: 7 seconds	

Experiment No 37

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.

Source Code

book_simple.py

```
class Publisher:

    def __init__(self, publisher):

        self.publisher = publisher

    def display(self):

        print("Publisher name:", self.publisher)

class Book(Publisher):

    def __init__(self, title, author):

        self.title = title

        self.author = author

    def display(self):

        super().display()

        print("Title of the book :", self.title)

        print("Author of the book:", self.author)

class Python(Book):

    def __init__(self, pub, author, title, price, no):

        self.price = price
```



```

        self.no_of_pages = no

    Book.__init__(self, title, author)

    Publisher.__init__(self, pub)

def display(self):

    super().display()

    print("Price of the book:", self.price)

    print("Number of pages in the book", self-
.no_of_pages)

b1 = Python("SPD", "Brian Jones ", "Python Cookbook:
Recipes For Mastering Python", 1670, 1230)

b1.display()

```

OUTPUT

Publisher name: SPD

Title of the book : Python Cookbook: Recipes For
Mastering Python

Author of the book: Brian Jones

Price of the book: 1670

Number of pages in the book 1230

Experiment No 38

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

Source Code

1 file.py

```
l = list()

f = open("demo.txt", "w")

n = int(input("Enter the number of lines:"))

for i in range(n):

    f.write(input("Enter some text:")+"\n")

f.close()

f = open("demo.txt", "r")

for i in f:

    print(i)

    l.append(i[:-1])

f.close()

print(l)
```

OUTPUT

Enter the number of lines: 5

Enter some text:line 1

Enter some text:line 2

Enter some text:line 3

Enter some text:line 4

Enter some text:line 5

line 1

line 2

line 3

line 4

line 5

['line 1', 'line 2', 'line 3', 'line 4', 'line 5']

demo.txt

line 1

line 2

line 3

line 4

line 5

Experiment No 39

Python program to copy odd lines of one file to other

Source Code

2_copy_odd_lines.py

```
f1 = open("demo1.txt", "w")
f2 = open("demo2.txt", "w")
n = int(input("Enter the number of lines:"))
for i in range(n):
    f1.write(input("Enter some text:")+"\n")
f1.close()
f1 = open("demo1.txt", "r")
count = 1
for i in f1:
    if count % 2 == 0:
        count += 1
        continue
    f2.write(i)
    count += 1
f1.close()
f2.close()
```

OUTPUT

Enter the number of lines: 2

Enter some text: winterfell

Enter some text: kings landing

demo1.txt

winterfell
kings landing

demo2.txt

winterfell

Experiment No 40

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

Source Code

3_read_csv.py

```
import csv

lst = []

with open("city.csv", "w") as file:

    write = csv.writer(file)

    write.writerow(["id", "place", "district"])

    limit = int(input("Enter the lines of data you want to enter:"))

    for i in range(limit):

        row_string = input("Enter data (id,place,district) separated by comma:")

        row_list = row_string.split(",")

        write.writerow(row_list)

with open("city.csv", "r") as file:

    read = csv.reader(file)

    for row in read:

        print(row)

        lst.append(",".join(row))

print(lst)
```

OUTPUT

Enter the lines of data you want to enter: 3

Enter data (id,place,district) separated by comma:55,vadakara,kozhikode

Enter data (id,place,district) separated by comma:44 ,nedumangad,trivandrum

Enter data (id,place,district) separated by comma:22,melechova ,kannur

```
['id', 'place', 'district']
```

```
['55', 'vadakara', 'kozhikode']
```

```
['44 ', 'nedumangad', 'trivandrum']
```

```
['22', 'melechova ', 'kannur']
```

```
['id,place,district', '55,vadakara,kozhikode', '44 ,ne-  
dumangad,trivandrum', '22,melechova ,kannur']
```

city.csv

```
id,place,district
```

```
55,vadakara,kozhikode
```

```
44 ,nedumangad,trivandrum
```

```
22,melechova ,kannur
```


Experiment No 41

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

Source Code

4_read_specific_column_csv.py

```
import csv

header = ["place", "name", "age"]

with open("city.csv", "w") as file:

    write = csv.DictWriter(file, fieldnames=header)

    write.writeheader()

    write.writerow({"place": "vatakara", "name":
"Samuel", "age": 21})

    write.writerow({"place": "kainatty", "name":
"Aswanth", "age": 21})

    write.writerow({"place": "Tholikkode", "name": "Ro-
jin", "age": 23})

    write.writerow({"place": "Palakkaadu", "name":
"Aleena", "age": 13})

with open("city.csv", "r") as file:

    read = csv.DictReader(file);

    n = input("Enter the column name you
want(place,name,age):")

    for i in read:

        print(i[n])
```

OUTPUT

Enter the column name you
want(place,name,age):place

Winterfell
Bravos
Beyond wall
Sydney

city.csv

place,name,age
Winterfell,Sansa,21
Bravos,Arya,21
Beyond wall,Robert,43
Sydney,Alderson,33

Experiment No 42

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

Source Code

5 dict to csv.py

```
import csv

header=["place","name","age"]

with open("city.csv","w") as file:

    write=csv.DictWriter(file,fieldnames=header)

    write.writeheader()

    limit = int(input("Enter the No.of lines you want
to enter:"))

    for i in range(limit):

        row_str = input("Enter the data in order
(place,name,age) separated by comma:")

        row_lst = row_str.split(",")

write.writerow({"place":row_lst[0],"name":row_lst[1],"a
ge":row_lst[1]})

with open("city.csv", 'r') as file:

    read=csv.DictReader(file);

    for i in read:

        print(dict(i))
```

OUTPUT

```
Enter the No.of lines you want to enter: 2
Enter the data in order (place,name,age) separated by
comma: Bravos ,Arya ,18
Enter the data in order (place,name,age) separated by
comma: Winterfell, Sansa ,21
{'place': ' Bravos ', 'name': 'Arya ', 'age': 'Arya '}
{'place': ' Winterfell', 'name': ' Sansa ', 'age': '
Sansa '}
```

city.csv

```
place,name,age
Bravos ,Arya ,Arya
Winterfell, Sansa , Sansa
```