

Date 3/02/2021

## Program No. 22

Aim: Find the sum of all numbers in a list.

Sum = 0

Input-String = Input("Enter a list of elements separated by space.")

PrintC("calculating sum of elements of input list")

Sum = 0

for num in list:

    Sum += int(num)

Print("sum = ", sum)

Result: The program has been executed and the output was verified.

output

is 60 elements

Enter a list of elements separated by space:

10 20 30

Calculating sum of elements of input list.

Sum = 60.

Input list length is 3.  
Input list has 3 elements.  
First element is 10.  
Second element is 20.  
Third element is 30.

Input list length is 3. (list)  
Input list has 3 elements.

First element is 10.  
Second element is 20.  
Third element is 30.

Input list length is 3. (list)  
Input list has 3 elements.

First element is 10.  
Second element is 20.  
Third element is 30.

Output length is 3.

(list) output length

Input list length is 3. (list)  
Input list has 3 elements.  
First element is 10.  
Second element is 20.  
Third element is 30.

Date 3/06/2011

### Program 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.

Proposed code:

```
num = list()
```

```
for i in range(1000, 10000):
```

```
    if (math.sqrt(i)) - (math.sqrt(i)) == 0:
```

```
        j = i
```

```
        while j > 1:
```

```
            x = int(j % 10)
```

```
            if (x % 2) == 0:
```

```
                break
```

```
j = int(j / 10)
```

```
if (j == 0):
```

```
    num.append(i)
```

```
print(num)
```

Result: The program has been executed and -the output was verified.

output

[4624, 6084, 6450, 8464]

Program No: 24

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

```

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

Rows = int(input("Enter the number of rows:"))

for i in range(1, Rows+1):

    for j in range(1, i+1)

#rows and column

        print(i\*j, end=" ")

    print()

Result : Program has been executed and the output was verified

Output

1

2 4

3 6 9

4 8 12 16.

Program No: 25

Aim: Count the number of characters in a string  
 (Character Frequency).

```
def char_frequency(sle):
    dict = {}
    for n in sle:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict.
```

```
sle = input("Enter string")
print(Chare_frequency(sle))
```

Result: Program has been executed and -the output can be verified.

Output

enter string asha

{'a': 2, 's': 1, 'h': 1}

## Program No: 26.

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

```
def Verbing(s):
    if len(s) < 3:
        print("Please enter a valid string")
    elif s.endswith('ing'):
        s = s + 'ly'
        print(s)
    else:
        s = s + 'ing'
        print(s)
    st = input("enter the string")
    Verbing(st)
```

Result: Program has been executed and the output can be verified.

output

enter the string data.

string

## Program No: 27

Aim: Generate all factors of a number.

```
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)
```

Result: The program has been executed and the output was verified.

Output

Enter the number 10

The factors of 10 are:

1  
2  
5  
10.

Program No. 28.

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

import math

R-area = lambda h, b: 1/2 \* b \* h

S-area = lambda s: s \* s

Print("triangle")

h = float(input("enter height of the triangle"))

b = float(input("enter base"))

Print("area =", R-area(b, h))

Print("square")

S = float(input("enter side"))

Print("area =", S-area(s))

Result: Program has been executed, and the output was verified.

Output

triangle

enter height of the triangle &

enter base &

Area = 8.0

Square

enter side

Area = 4.0

## Program : 29.

Aim: Python program to 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 to find area and perimeter of figures by different importing statements.

### Circle.py

```
def area(r):
    print('Area of circle with radius:', r, 'is',
          'area': 1. (3.14 * r * r), 'units')

def circumference(c):
    print('circumference of circle with radius:',
          r, 'is:', 'circumference': 2 * (3.14 * r), 'units')
```

### Rectangle.py

```
def area(a,b):
    print('Area of rectangle with sides', a,
          'and', b, 'is': 'area': 1. (a*b), 'sq.units')
```

Find perimeter.py

import circle.

from rectangle import \*

from Graphics.3D-graphics import Cuboid,Sphere.

a = float(input('Enter length of the rectangle:'))

b = float(input('Enter breadth of the rectangle:'))

rectangle.perimeter(a,b)

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

circle.circumference(r)

f = float(input('Enter length of the cuboid:'))

b = float(input('Enter breadth of the cuboid:'))

h = float(input('Enter height of the cuboid:'))

cuboid.perimeter(f,b,h)

R = float(input('Enter the radius of the sphere:'))

Sphere.perimeter(R)

Find Area.Py

import circle.

from rectangle import \*

from Graphics.3D-graphics import Cuboid,Sphere.

def perimeter(a,b):

Print('Perimeter of rectangle with sides,'  
a, 'and', b, 'is:', 'def' 'd=(2\*(a+b))', 'units')

Sphere.py

def area(r):

Print('Area of sphere with radius:', r, 'is:', ' $\pi r^2$ ', '+', '(4 \* (3.14 \* r \* r))', 'sq.units')

def perimeter(r):

Print('Perimeter of (great circle of) sphere  
with radius:', r, 'is:', 'def' 'p' '+', '(2 \* 3.14 \* r)', ', units')

Cuboid.py

def area(l,b,h):

Print('Total surface area of cuboid with  
dimensions', l, 'x', b, 'x', h, 'is:', 'def' 's' '+', '(2 \*  
(l \* b + b \* h + l \* h))', 'sq.units')

def perimeter(l,b,h):

Print('Perimeter of cuboid with dimensions',  
l, 'x', b, 'x', h, 'is:', 'def' 'p' '+', '(4 \*  
(l + b + h))', 'units').

```
a = float(input('Enter length of the rectangle :'))
```

```
b = float(input('Enter breadth of the rectangle :'))
```

```
area(a,b)
```

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

```
circle.area(r)
```

```
l = float(input('Enter length of the cuboid '))
```

```
b = float(input('Enter breadth of the cuboid '))
```

```
h = float(input('Enter height of the cuboid '))
```

```
cuboid.area(l,b,h)
```

```
o = float(input('Enter the radius of sphere :'))
```

```
sphere.area(o)
```

## Result

The program has been executed and output was verified.

## Output

Enter length of the rectangle : 4

Enter length of the rectangle : 3

Perimeter of the rectangle with side 4.0 and 3.0  
is: 14.00 units

Enter -the radius of the circle : 2

Circumference of circle with radius 2.0 is 12.56  
units

Enter length of the cuboid : 5

Enter breadth of the cuboid : 4

Enter height of the cuboid : 3

Perimeter of cuboid with dimensions 5.0, 4.0, 3.0,  
is 48.00 units.

Enter -the radius of the sphere : 2

Perimeter of sphere with radius 2.0 is  
12.56 units.

Date 17/09/2021

Program No: 30.

Aim: Python program to create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two rectangle objects by their area.

class Rectangle

def \_\_init\_\_(self, l, b):

self.length = l

self.breadth = b

def area(self):

return self.length \* self.breadth

def perimeter(self):

return 2 \* (self.length + self.breadth)

def Cmp(self, obj):

If self.area() > obj.area():

Print('Rectangle with length = ', self.length, 'and' ' breadth = ', self.breadth, 'has the greater area')

elif self.area() < obj.area():

Print('Rectangle with length = ', obj.length, 'and' ' breadth = ', obj.breadth, 'has the greater area')

• length, (and breadth =; obj.breadth, has  
the greater area.)  
else:

Point C: They have equal area.)

21. Rectangle(2,3)

22. Rectangle(3,4)

21. cmp(24)

Result: The program has been executed and the  
output was verified.

Output

Rectangle with length = 9 and breadth = 3 has  
the greater area.

## Program No. 31

Aim: Python program - to create a Bank account with members account number, name, type of account and balance. Write constructors and methods - to deposit at the bank and withdraw an amount from the bank.

class BankAccount:

def \_\_init\_\_(self, a, n, t, b):

self.acno = a

self.name = n

self.type = t

self.bal = b

def deposit(self, a):

self.bal += a

Print('Rs', a, 'deposited', current balance  
is Rs', self.bal)

def withdraw(self, a):

if self.bal >= a:

self.bal -= a

Print('Rs', a, 'withdrawn', current balance  
is Rs', self.bal)

else:

Print ('Insufficient balance to make this transaction')

a = int(input('Enter account number:'))

n = int(input('Enter name of the account holder:'))

t = int(input('Enter account type'))

b = int(input

b = float(input('Enter your balance:'))

aci = BankAccount(a, n, t, b)

aci.deposit(float(input('Enter amount to deposit:')))

aci.withdraw(float(input('Enter amount to withdraw')))

Result: Program has been executed and the output was verified.

## Output

Enter account number : 0090210909090909

Enter name of the account holder : John.

Enter account type : Savings

Enter your balance : 100000

Enter amount to deposit : 300000

Rs 300000 deposited! Current balance is

Rs 400000.0

Enter amount to withdraw : 5000

Rs 5000.0 withdrawn! Current balance is

395000.0

## Program No. 32.

Aim: Python program to create a class rectangle with private attribute length and width

Overload ' $<$ ' operator to compare the area of 2 rectangles.

Class Rectangle:

```
def __init__(self, l, w):
```

```
    self.length = l
```

```
    self.width = w
```

```
    self.area = self.width * self.length.
```

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

```
    if self.area < other.area:
```

```
        print('Rectangle with length=', self.length,  
              'and width=', self.width,  
              'has the lesser area!')
```

```
    elif other.area < self.area:
```

```
        print('Rectangle with length= ', other.length,  
              'and width= ', other.width,  
              ', has the lesser area!')
```

else

```
    print('They have equal area!')
```

$f = \text{float}(\text{input}(\text{'Enter length of 1st rectangle:')}))$

$w = \text{float}(\text{input}(\text{'Enter width of 1st rectangle'}))$

$R_1 = \text{Rectangle}(l, w)$

$l = \text{float}(\text{input}(\text{'Enter length of 2nd rectangle:'}))$

$w = \text{float}(\text{input}(\text{'Enter width of 2nd rectangle'}))$

$R_2 = \text{Rectangle}(l, w)$

$R_1 < R_2$ .

Result: Program has been executed and output  
was verified.

## Output

Enter length of 1st rectangle : 7

Enter width of 1st rectangle : 8

Enter length of 2nd rectangle : 8

Enter width of 2nd rectangle : 7

They have equal area.

## Program No: 33

Aim: Python program to display future leap years from current year to a final year entered by user.

```

year = int(input("Enter initial year"))
Year1 = int(input("Enter final year"))
count = year
while count <= Year1:
    if year % 4 == 0 and (year % 100 != 0 or
                           year % 400 == 0):
        print(year)
    year = year + 4
    count = year.

```

Result: Program has been executed and output was verified.

output

Enter initial year: 2000

Enter final year: 2020.

2000

2004

2005

2012

2016

2020

## Program No.34

Aim: Python program to create a list of colours from comma separated colors entered by user  
 Display first and last colours.

Input- String = input ("Enter a list element separated by comma")

list = input\_string.split (",")

Print ("The entered list is:")

for x in range (len(list)):

Print (list [x])

Print ("The first and last colour is \n")

Print ("Ans is " + list [0] + list [-1])

Result: Program has been executed and output was verified.

## Output

Enter colors separated commas red black yellow

The entered list is red, black, yellow.

The first and last color is red yellow.

Date : 13/04/2021

Program No: 35

Aim: Python program to print all colors from color list which are not contained in color list.

```
colors1 = set ((input('Enter colors separated by commas: ')).split(',')))
```

```
colors2 = set ((input('Enter colors separated by commas: ')).split(',')))
```

Print ('Colors in color-list, not contained in color-list are:', list (colors1.difference(colors2))))

Result: Program has been executed and the output was verified.

Output

Enter colors separated by commas: red, yellow,  
brown.

Enter colors separated by commas: black, white.  
colors in color list are not contained in color-list  
are ['brown', 'red', 'yellow']

Program No. 32

From Python program to C code positive list of integers from given list of integers.

list1 = [1, -4, 2, -3, 5, -2, -43, 37, -375, -756, 244  
-66, 7]

Pos = list()

for i in list1:

if i > 0:

Pos.append(i)

Print("Original list : ", list1)

Print("Positive integer list : Pos")

Result: Program executed and the output was verified.

## Output

Original list: [1, -4, 2, -3, 8, -2, -43, 37,  
-378, -756, 244, -667]

Positive integer list: [1, 2, 8, 37, 244]

## Program No: 37

Aim: Python program to find biggest of 3 numbers entered.

```

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

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

```

Print("The largest number is:", large).

Result: Program executed and the output was verified.

## Output

Enter first number : 9

Enter second number : 10

Enter third number : 11.

The largest number is 11.0.

## Program No. 38

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

```
def chaes-mix-up(a,b):
    new_a = b[:2] + a[2:]
    new_b = a[:2] + b[2:]
    return new_a + " " + new_b
```

```
a = input("Enter first string")
b = input("Enter second string")
print(chaes-mix-up(a,b))
```

Result: Program has been executed and the output was Verified.

## Output

enter first string hello.  
enter second string welcome.  
Hello welcome.

(Connection closed by peer) (Connection closed by peer)  
(Connection closed by peer) (Connection closed by peer)  
(Connection closed by peer) (Connection closed by peer)

connection closed  
Good bye

Connection closed by peer

## Program No 29

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

Class Time:

def \_\_init\_\_(self, hh=0, mm=0, ss=0):

Self.hour = hh.

Self.minute = mm.

Self.second = ss.

def \_\_add\_\_(self, other):

second = int((self.second + other.second) % 60)

minute = int(((self.minute + other.minute) \* 60 + (self.second + other.second)) / 60)

hour = int(((self.hour + other.hour) \* 24 + (self.minute + other.minute)) / 60)

print('Time[hh:mm:ss]', hour, ':', minute, ' ', second)

T1 = Time(10, 20, 33)

T2 = Time(16, 15, 6)

$T_1 + T_2$

Result: Program has been executed and output was  
verified.

## Output

Time [hh:mm:ss] 2:35:41

## Program No: 40

Aim: Create a class publisher(name). Derive class book from publisher with attributes book and author. Derive class python from book with attributes price and no.of.pages. Write a program that displays information about python book. Use base class constructor invocation and method overriding.

```

class Publisher:
    def __init__(self, name):
        self.name = name
    def show(self):
        pass

class Book(Publisher):
    def __init__(self, title, author, name):
        self.title = title
        self.author = author
        Publisher.__init__(self, name)
    def show(self):
        pass

```

```
class Python(Book):  
    def __init__(self, p, no, title, author, name):  
        self.price = p  
        self.no.of.pages = no
```

```
Book.__init__(self, title, author, name)  
    def show(self):  
        print('Book title:', self.title)  
        print('Author:', self.author)  
        print('Author publisher:', self.name)  
        print('Price:', self.price)  
        print('No of pages', self.no.of.page)
```

```
P1 = Python(250.90, 250, 'Programming in C',  
            'Balaguruswami', 'ABC Books')  
P1.show()
```

Result: Program has been executed and output  
was verified.

## Output

Book title : Programming in C

Author : Balaguruswami

Publisher : ABC Books

Price : ₹ 6. 250. 9

No. of pages : 250.