

Program no: 1

Aim: python program to find the square of a number

Program:

```
>>> a = int(input("Enter the Integer"))
```

Enter the Integer 5

```
>>> square = a * a
```

```
>>> print(f"Square of {a} is {square}")
```

Square of 5 is 25

```
>>>
```

Output

Square of 5 is 25

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

Program no: 2

Aim: Python program to return the area of a circle  
using the function.

Program:

```
>>> def area(r):
    return 3.14 * r * r;
>>> n = float(input("Enter radius"))
Enter radius 5
>>> print("area = %.6f" % area(n))
Area = 78.500000
>>>
```

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

Output:  
area = 78.500000

Program no: 3

Aim: Python programme to find the largest among the three numbers given by the user

Program:

```
>>> a = float(input("enter 1st number; "))

enter 1st number; 5

>>> b = float(input("enter 2nd number; "))

enter 2nd number; 6

>>> c = float(input("enter 3rd number; "))

enter 3rd number; 9

>> if (a > b):

    if (b > c):

        i = a

    else:

        if (c > a):

            i = c

        else:

            i = a

else:

    if (a > c):

        i = b

    else:

        if (c > b):

            i = c

        else:

            i = b

>> print("largest number is", i)

largest number is 9.0
```

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

Output

Largest number is 9.0

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

Program:

```
>>> print("Enter year")
Enter year
>>> endYear = int(input())
2030
>>> startYear = 2020
>>> print("List of leap years:")
List of leap years:
>>> for year in range(startYear, endYear):
    if (0 == year % 4) and (0 != year / 100) or (0 == year % 400):
        print(year)
```

2020

2024

2028

>>>

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

Output

2020

2024

2028

Output

```
14 Squared is 196  
21 Squared is 441  
11 Squared is 121  
9 Squared is 81  
7 Squared is 49  
5 Squared is 25
```

Output:

The vowels present in the string:

```
{'o', 'i', 'u', 'a', 'e'}
```

Program no: col1:(3b) : 5

Aim: Square of N numbers

Program:

```
>>> list1 = {14, 21, 11, 9, 7, 5}  
>>> for n in list:  
    Square = n**2  
    print(n, 'squared is', Square)
```

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

Program no: col1:3c

Aim: form a list of vowels selected from given word.

Program:

```
>>> string A = "Hi! how are you"  
>>> print("Given string: \n", string A)  
Given string:  
Hi! how are you.  
>>> vowels = A[0:1:2:10]  
>>> res = set([each for each in string A if each in vowels])  
>>> print("The vowels present in the string:\n", res)
```

Result: The program has been executed and output was verified

Program no: CO1-(4) : 6

Aim: Count the occurrence of each word in line of text

Program:

```
>>> def word_count(st):  
    counts = dict()  
    words = st.r.split()
```

```
    for word in words:
```

```
        if word in counts:
```

```
            counts[word] += 1
```

```
        else:
```

```
            counts[word] = 1
```

```
    return counts
```

```
>>> print(word_count('the quick brown fox jumps over the  
lazy dog.'))
```

Output

```
{'the': 2, 'quick': 1, 'brown': 1, 'fox': 1, 'jumps': 1, 'over': 1,
```

```
'lazy': 1, 'dog': 1}
```

```
>>>
```

Result: The program has been executed and output has been verified

output

Count of a in the list is : 5

>>>

Program no : Col : (6) : 7

Aim : Store a list of first names. Counts the occurrence of 'a' within the list

Program :

```
>>> a = ['adarsh', 'ram', 'rohan', 's']
```

```
>>> str1 = (' '.join(a))
```

```
>>> count = 0
```

```
>>> for i in str1:
```

```
    if i == 'a':
```

```
        count = count + 1
```

```
>>> print ("Count of a in the list is :" + str(count))
```

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

Output

the lengths are not equal.

Program no: Col : 7 (a) : 8

Aim: Enter 2 lists of integers. check (a) whether list are of same length.

Program:

```
>>> list 1 = [10, 11, 12, 12, 13, 14, 15, 6, 18]
>>> list 2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]
>>> l1 = len(list 1)
>>> l2 = len(list 2)
>>> if l1 == l2:
    print("both have same length")
else:
    print("the lengths are not equal")
```

Result: The program has been executed and output was Verified

Program no: Col : 7 (b)

Aim: whether list sum to same value

Program:

```
>>> list 1 = [10, 11, 12, 12, 13, 14, 15, 6, 12]
>>> list 2 = [16, 12, 13, 14, 15, 16, 10, 11, 11]
>>> s1 = sum(list 1)
>>> s2 = sum(list 2)
```

output

The sum of list are not equal.

output

The sum of list are equal

Program no: col : (f b) : 9

Aim: whether list sum to same value

Program:

```
>>> list 1 = [10, 11, 12, 12, 13, 14, 15, 6, 12]  
>>> list 2 = [16, 12, 13, 14, 15, 16, 10, 11, 1]  
>>> S1 = sum (list1)  
>>> S2 = sum (list2)  
>>> if (S1 == S2)
```

```
    print ("The sum of list are equal")
```

```
else:
```

```
    print ("The sum of the list are not equal")
```

```
>>> list 1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]
```

```
>>> list 2 = [16, 12, 13, 14, 15, 16, 16, 11, 12, 16, 12]
```

```
>>> S1 = sum (list1)
```

```
>>> S2 = sum (list2)
```

```
>>> if (S1 == S2)
```

```
    print ("The sum of list are equal")
```

```
else:
```

```
    print ("The sum of the list are not equal")
```

Result: The program has been executed and output was verified

Program no: E01 : Q3) : 10

Aim: whether any value occur in both

program:

```
>>> list1 = [10, 11, 12, 13, 14, 15, 6, 12]
>>> list2 = [16, 18, 13, 14, 15, 16, 10, 11]
>>> for ele in list1:
    if ele in list2:
        flag = 0
>>> if (flag == 0):
    print("common elements exist")
else:
    print("No common elements")
```

Output

Common element exist

Output

Common element doesn't exist

```
>>> list1 = [10, 11, 12, 13, 14, 15, 6, 12]
```

```
>>> list2 = [1, 2, 3, 4, 5]
```

```
>>> for ele in list1:
    if ele in list2:
        flag = 1
```

```
>>> if (flag == 1):
    print("common elements exist")
```

```
else:
```

```
    print("common elements doesn't exist")
```

Result: The program has been executed and output was verified

Output

Picku\$

program no: CO1 : (8) : 11

Aim: Get a string from a input string where all occurrence  
of first character replaced with '\$', except first character

Program:

```
>>> def change_char(str1):  
    char = str1[0]  
    str1 = str1.replace(char, '$')  
    str1 = char + str1[1:]  
    return str1  
>>> print(change_char('pickup'))
```

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

Program no: Col : (9) : 12

Aim: Create a string from given string where first last character exchanged

Program:

```
>>> def change_string(str):  
    return str[-1:] + str[1:-1] + str[:1]  
>>> print(change_string('AMALJIYOTHI'))
```

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

Output

AMALJIYOTHI

Program no: CO1 (10): 13

Aim: Accept the radius from user and find area of circle

Program:

```
>>> from math import pi  
>>> r = float (input ("Input the radius of the circle :"))  
Input the radius of the circle : 4  
>>> print ("The area of the circle with radius " + str(r)  
is : " + str(pi * r**2))
```

T

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

Output

The area of the circle with radius 4.0 is:

50.26548245743669

output

861

Program no: (01 : (4) : 14

Aim: Accept an integer n and compute n!+n!+n!

Program:

```
>>> a = int(input("Input an integer :"))  
Input an integer :7  
>>> n1 = int("%s" % (a))  
>>> n2 = int("%s%s" % (a,a))  
>>> n3 = int("%s%s%s" % (a,a,a))  
>>> print(n1+n2+n3)
```

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

Output

Dictionary in ascending order by value:

[0,0),(2,1),(1,2),(4,3),(3,4)]

Output

Dictionary in descending order by value:

{3:4, 4:3, 1:2, 2:1, 0:0}

program no.: 01:(7):15-

aim : Sort dictionary in ascending and descending order

Program:

```
>>> import operator
```

```
>>> d = {1:2, 3:4, 4:3, 2:1, 0:0}
```

```
>>> print ('ordinary dictionary : ', d)
```

Original dictionary : {1:2, 3:4, 4:3, 2:1, 0:0}

```
>>> sorted_d = sorted(d.items(), key=operator.itemgetter(1))
```

```
>>> print ('Dictionary in ascending order by value : ', sorted_d)
```

Dictionary in ascending order by value : [(0,0), (2,1), (1,2), (4,3), (3,4)]

```
>>> sorted_d = dict(sorted(d.items(), key=operator.itemgetter(1),
```

reverse=True))

```
>>> print ('Dictionary in descending order by value : ', sorted_d)
```

Dictionary in descending order by value : {3:4, 4:3, 1:2, 2:1, 0:0}

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

Program no: (01 :18): 16  
Aim: Merge two dictionaries

Program:

```
>>> def merge(dict1, dict2):  
    return (dict2, update(dict1))  
  
>>> dict1 = {'a': 10, 'b': 8}  
>>> dict2 = {'d': 6, 'c': 4}  
>>> print(merge(dict1, dict2))  
None  
>>> print(dict2)  
{'d': 6, 'c': 4, 'a': 10, 'b': 8}
```

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

Output

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

program no: col : (9) : 17  
Aim: find gcd of 2 numbers  
program:

```
>>> def gcd(a,b):  
    # Everything divides 0  
    if (b==0):  
        return a  
    return gcd(b, a%b)  
>>> a=98  
>>> b=56  
>>> if (gcd(a,b)):  
    print('GCD of', a, 'and', b, 'is', gcd(a,b))  
else:  
    print('not found')
```

output

GCD of 98 and 56 is 14

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

Output

[11, 33, 55]

Program no: (01 : t80) : 18

Aim: From a list of integers, create a list removing even numbers

Program:

```
>>> list = [11, 12, 33, 44, 55]
```

```
>>> print(list)
```

```
[11, 12, 33, 44, 55]
```

```
>>> for i in list:
```

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

```
        list.remove(i)
```

```
>>> print("list after removing Even numbers:")
```

```
list after removing Even numbers:
```

```
>>> print(list)
```

```
[11, 33, 55]
```

```
>>>
```

Result: The program has been executed and output was verified

output

Enter the number for which Factorial is needed

5

120

Program no: CO2:Q1:19

Aim: program to find factorial of a number

Program:

```
print ("Enter the number for which factorial is needed")
x = int (input ())
f = 1
if x < 0:
    print ('factorial for -ve doesn't exist')
elif x == 0:
    print ('factorial of 0 is 1')
else:
    for i in range (1, x+1):
        f = f * i
    print (f, "\n")
```

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

Program no: 18): 20

Aim: Generate Fibonacci Series of N terms

Program:

```
x = int(input("Enter the limit for the Fibonacci"))
p1 = 0
p2 = 1
if (x < 0):
    print("Fibonacci series doesn't exist for this input:")
else if x == 1:
    print(p1)
else if x == 2:
    print(p1)
    print(p2)
else:
    print(p1)
    print(p2)
for i in range(1, x):
    p3 = p1 + p2
    p1 = p2
    p2 = p3
    print(p3)
```

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

Output

Enter the limit for the Fibonacci

0  
1  
1  
2  
3  
5

Output

21

Process finished with exit code 0

Program no: (02:3): 21

Aim: find the sum of all items in a list

Program:

```
def sum_list(items):
    sum_numbers = 0
    for x in items:
        sum_numbers += x
    return sum_numbers
print(sum_list([9, -4, 6, 10]))
```

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

Program no: 4 (02 : 14) : 22

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

Program:

import math

for x in range(1000, 10000):

    num = str(x)

    c = 0

    for y in range(4):

        n = int(num[y])

        if n % 2 == 0

            c += 1

        if c == 4:

            root = math.sqrt(x)

            if int(root + 0.5) \*\* 2 == x:

                print(x)

• output

4624

6084

6400

8464

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

Program no: Co2:(5):R3

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

1  
2 4  
3 6 9  
4 8 12 16

Program:

```
X = int (input ("Enter the number"))  
for i in range (1, X+1):  
    p = i  
    for j in range (1, i+1):  
        p = i * j  
        print (p, end = ' ', flush = True)  
    print (' ', end = ' ', )  
print ("\n")
```

Output

Enter the number

1  
2 4  
3 6 9  
4 8 12 16  
5 10 15 20 25

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

Program no: Co2 : (6) : 2.4

Aim: Count the number of characters (character-frequency) in a string

Program:

```
def char_frequency(stri):
```

```
    dict = {}
```

```
    for n in stri:
```

```
        key = dict.keys()
```

```
        if n in keys:
```

```
            dict[n] += 1
```

```
        else:
```

```
            dict[n] = 1
```

```
    return dict
```

```
print(char_frequency("I 'm BATMAN"))
```

Output

```
{'I':1, "'":1, 'M':2, ' ':1, 'B':1, 'A':2, 'T':1, 'N':1}
```

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

Program no: [02 : 7] = 25

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

Program:

```
str1 = str(input("Enter the string:"))
x = len(str1)
if (x >= 3):
    if (x < 3):
        str2 = str1 + "ing"
    else:
        if str1[-3:] == "ing":
            str2 = str1 + "ly"
        else:
            str2 = str1 + "ing"
print(str2)
```

### Output

Enter the string: Joking

Jokingly

Enter the string: Jok

Joking

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

Program no : C02 T(8) : 26

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

Program :

```
def func(strings, y):  
    g = len(strings[0])  
    for i in range(y):  
        if (g < len(strings[i])):  
            g = len(strings[i])  
    return g
```

```
input_string = input ("Enter a list element separated by space")  
list = input_string.split()  
y = len(list)  
print(func(list, y))
```

Output

Enter a list element separated by space  
Alice Amaljyothi Kottayam

10

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

Program no : CO2:(9):27

Aim : Construct following pattern using nested loop

Program:

```
for i in range (1,10):
    if i<=5:
        for y in range (i):
            print("+", end=' ', flush=True)
        print("\n")
    else:
        for y in range (i,10):
            print("+", end=' ', flush=True)
        print("\n")
```

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

Output

.....  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Program no: C02:09:28

Aim: generate all factors of a number

Program:

```
def print_factors(x):
```

```
    print ("The factors of", x, "are:")
```

```
    for i in range (1,x+1):
```

```
        if x % i == 0:
```

```
            print(i)
```

```
x = int (input ("Enter the number whose factors are to  
be found"))
```

```
print_factors(x)
```

### Output

The factors of 256 are:

1  
2  
4  
8  
16  
32  
64  
128  
256

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

Program no : CO2 : (1) : 29

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

program:

```
s = int(input("Enter the length of side of square:"))
l = int(input("Enter the length of rectangle:"))
b = int(input("Enter the breadth of rectangle:"))
print('Enter the base and height of triangle')
h = int(input("Enter the base of triangle:"))
cl = int(input("Enter the height of triangle:"))
x = lambda s: s * s
y = lambda l, b: l * b
t = 0.5
z = lambda h, cl, t: h * cl * t
print("Area of square is:", x(s))
print("Area of rectangle", y(l, b))
print("Area of triangle", z(h, cl, t))
```

output

```
Enter the length of side of square: 2
Enter the length of rectangle: 2
Enter the breadth of rectangle: 2
Enter the base and height of triangle
Enter the base of triangle: 2
Enter the height of triangle: 2
Area of square is: 4
Area of rectangle 4
Area of triangle 2.0
```

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

program no: 03 : 2 : 35

Aim : create a package graphics with modules rectangle, circle and sub-package 3D graphics with modules cuboid & sphere, include methods - to find area and perimeter of respective figures in each modules, write programs that find area and parameter of figures by different importing statements.

Program :

### Circle.py

def area(r):

```
print('Area of circle with radius', r, 'is:', '%.2f' %  
(3.14 * r * r), 'sq.units')
```

def circumference(r):

```
print('circumference of circle with radius', r, 'is:', '%.2f'  
'%. (3.14 * 2 * r), 'units')
```

### rectangle.py

def area(a, b):

```
print('area of rectangle with sides', a, 'and', b, 'is:',  
'%.2f' % (a * b), 'sq.units')
```

def perimeter(a, b):

```
print('Perimeter of rectangle with sides', a, 'and', b,  
'is:', '%.2f' % (2 * (a + b)), 'units')
```

## Sphere.py

def area(r):

Print ('Area of sphere with radius', r, 'is:', '%.2f' %  
(4 \* (3.14 \* r \* r)), 'sq. units')

def penmeter(r):

Print ('Penmeter of great circle of sphere with radius',  
'r, is:', '%.2f' % (2 \* 3.14 \* r), 'units')

## Cuboid.py

def area(l, b, h):

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

def penmeter(l, b, h):

Print ('Penmeter of cuboid with dimensions',  
'l, b, h, is:', '=%.2f' % (4 \* (l + b + h)), 'units')

## findpenmeter.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:'))  
perimeter(a,b)  
  
r = float(input('Enter in the radius of the circle:'))  
circle_circumference(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_perimeter(l,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 import cuboid, sphere  
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)

```
r = float(input('Enter the radius of the sphere :'))  
sphere-area(r)
```

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

Output:

Enter length of the rectangle : 4

Enter breadth of the rectangle : 3

perimeter of rectangle with sides 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 48.0

Enter the radius of the sphere : 2

Parameter of (great circle of) sphere with radius

2.0 is 12.56 units.

Enter length of the rectangle : 2

Enter breadth of the rectangle : 3

Area of rectangle with sides 2.0 and 3.0 is 6.00 sq units

Enter the radius of the circle : 4

Area of circle with radius 4.0 is 50.24 sq units

Enter length of the cuboid : 4

Enter breadth of the cuboid : 7

Enter height of the cuboid : 2

Total surface area of cuboid with dimensions 4.0, 7.0, 2.0  
100.00 sq units

Enter the radius of the sphere : 1

Area of Sphere with radius 1.0 is 12.56 sq units

program no: 36

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.

Program:

```
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')
```

else:

    print ('They have equal area!')

    r1 = Rectangle (9, 3)

    r2 = Rectangle (3, 9)

    r1.Cmp (r2)

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: 37

Aim: Python program to create a Bank Account with members account number name, type & account and balance. write constructor and methods to deposit at the bank and withdraw an amount from the bank

Program:

Class Bank Account:

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('In sufficient balance to make this

Transactions!")

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

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

t = input('Enter account type:')

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

act1 = Bank Account(a, n, t, b)

act1.deposit(float(input('Enter amount to deposit:'))))

act1.withdraw(float(input('Enter amount to withdraw'))))

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

Output

Enter account number : 00900909090909

Enter name of the account holder : John

Enter account type : savings

Enter your balance : 100000

Enter amount to deposit : 30000

Rs 30000.0 deposited ! current balance is Rs  
40000.0

Enter amount to withdraw : 5000

Rs 5000.0 withdrawn ! current balance is,  
Rs 35000.0.

program no: 38

Aim : python program to create a class Rectangle with private attribute length and width overload '' operator to compare the area of 2 rectangles

Program :

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!')

else if other.area < self.area:

print('Rectangle with length = ', other.length,

'and width = ', other.width, 'has the  
lesser area!')

else:

print('They have equal area!')

l = float(input('Enter length of 1st rectangle:'))

w = float(input('Enter width of 1st rectangle:'))

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

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

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

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

$R_1 < R_2$

Output:

Enter length of 1st rectangle : 7

Enter breadth of 1st rectangle : 8

Enter length of 2nd rectangle : 8

Enter width of 2nd rectangle : 7

They have equal area!

Program no : 39

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

Program :

class publisher:

def \_\_init\_\_(self, 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 ('Publisher:', self.name)

print ('Price:', self.price)

print ('No of Pages:', self.no.of.pages)

P1 = Python (568.90, 250, "Programming with python,"  
"Guido Rossum", 'ABC Books')

P1.show()

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

Output:

Book title : Programming with Python

Author : Av. Rossen

Publisher : ABC Books

price : \$ 65.9

No. of pages: 250,

program no:40

Aim : Python programming to read a file line by line and store it into a list.

Program:

```
def file-read(fname):
    with open(fname) as f:
        # Content-list is the list that contains the
        # read lines
        c = f.read().splitlines()
        print(c)
        # print(len(c))
    file-read("demo.txt")
```

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

output: -

[A trailer is a vehicle designed for carrying bulk material, often on building sites. In, 'They are distinguished from dump trucks by configuration. I dumped']