

Output :-

Enter the value: 3

Area is 28.2600

Output:-

Enter the first number: 45

Enter the second number: 55

Enter the third number: 27

The largest number is 55

Date: 14-01-2021

Program No: 1

Aim: Python program to find area

```
def area(r):  
    pi = 3.14  
    return pi * (r * r);  
  
num = float(input("Enter the value: "))  
print("Area is %6f" % area(num));
```

Result: The program has been executed and output was verified

Date: 16-01-2021

Program No: 2

Aim: Python program to find largest among 3 numbers

```
num1 = float(input("Enter the first number: "))  
num2 = float(input("Enter the second number: "))  
num3 = float(input("Enter the third number: "))  
  
if (num1 > num2) and (num1 > num3)  
    largest = num1  
elif (num2 > num1) and (num2 > num3)  
    largest = num2  
else  
    largest = num3
```

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

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

Output :-

Enter an integer number : 5
Square of 5 is 25

Output :-

The radius of the circle : 4
The area of the circle with radius 4.0 is 50.2654

Output :-

14 squared is 196
20 squared is 400
13 squared is 169
8 squared is 64
6 squared is 36
2 squared is 4

Date: 16-01-2021

Program No: 3

Aim: Python program to find square of a number

```
num = int(input("Enter an integer number: "))  
square = num * num  
print("Square of {num} is {square}")  
print
```

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

Date: 26/01/2021

Program No: 4

Aim: Python program to find area of circle

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

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

Date: 26-01-2021

Program No: 5

Aim: Python program to find square of n

```
list1 = [14, 20, 13, 8, 6, 2]  
for n in list1:  
    square = n * n  
    print(n, squared is, square)
```

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

Output:-

Given string:

Hello.. How are you

The vowels present in the string

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

Output:-

{ 'python': 1, 'is': 1, 'a': 1, 'very': 1, 'versatile': 1, 'language': 1 }

Date: 26-01-2021

Program No: 6

Aim: Python program to find vowels in a string

```
stringA = "Hello.. How are you "
```

```
print ("Given string: \n", stringA)
```

```
vowels = "aAeEiIoOuu"
```

```
res = set([each for each in stringA if each in vowels])
```

```
print ('The vowels present in the string: \n', res)
```

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

Date: 26-01-2021

Program No: 7

Aim: Python program to count words in a sentence

```
def word_count(str):
```

```
    counts = dict()
```

```
    words = str.split()
```

```
    for word in words:
```

```
        if word in counts:
```

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

```
        else
```

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

```
    return counts
```

```
print(word_count('Python is a very versatile language'))
```

Result: The program executed and the output was verified.

Output:

Count of a in the list is: 5

Date: 26-01-2021

Program No: 8

Aim: Python program to count a in a list

```
a = ['anjali', 'reenu', 'gretta', 'aamy']
```

```
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.

Date: 26-01-2021

Program No: 9

Aim: Python program to check the length of list

```
list1 = [1, 2, 3, 4, 5, 6]
```

```
list2 = [9, 8, 7, 6, 3, 5]
```

```
len1 = len(list1)
```

```
len2 = len(list2)
```

```
if len1 == len2:
```

```
    print('both list have equal length')
```

```
else:
```

```
    print('both list doesn't have equal length')
```

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

Output:

both list have equal length

Output :

both list does not have equal sum

Output :

There are common elements

Date: 26-01-2021

Program No: 10

Aim: Python program to check the sum of list

```
list1 = [9, 4, 3, 7, 2]
```

```
list2 = [3, 4, 5, 8, 6]
```

```
total1 = sum(list1)
```

```
total2 = sum(list2)
```

```
if total1 == total2 :
```

```
    print('both list have equal sum')
```

```
else:
```

```
    print('both list does not have equal sum')
```

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

Date: 26-01-2021

Program No: 11

Aim: Python program to check the common elements in the list

```
list1 = [2, 3, 7, 5, 6, 9]
```

```
list2 = [8, 2, 3, 4, 5, 6]
```

```
for value in list1:
```

```
    if value in list2:
```

```
        common = 1
```

```
if common == 1:
```

```
    print('There are common elements')
```

```
else:
```

```
    print('No elements are common')
```

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

Output:

Oni\$n

Output:

nythop

Date: 27-01-2021

Program No: 12

Aim: Python program to replace a character

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

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

Date: 27-01-2021

Program No: 13

Aim: Python program to exchange the first and last letter in a string

```
def change_string(str1):  
    return str1[-1:] + str1[1:-1] + str1[:1]  
print(change_string('python'))
```

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

Output :-

None

{'d': 3, 'e': 2, 'l': 6, 'g': 5}

Output :

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

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

Dictionary in descending order by value : {5:4, 2:3, 1:2, 0:1, 4:0}

Date: 27-01-2021

Program No: 14

Aim: Python program to merge 2 dictionaries

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

dict1 = {'l': 6, 'g': 5}

dict2 = {'d': 3, 'e': 2}

print(Merge(dict1, dict2))

print(dict2)

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

Date: 27-01-2021

Program No: 15

Aim: Python program to ascent and descent dictionary

import operator

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

print('original dictionary:', d)

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

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

sorted_d = dict(sorted(d.items(), key=operator.itemgetter(1), reverse=True))

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

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

Output:

[10, 13, 26, 29, 38, 50]

list after removing even numbers:

[13, 29]

Output:

GCD of 144 and 12 is 12

Date: 27-01-2021

Program No: 16

Aim: Python program to remove even numbers from the list

```
list = [10, 13, 26, 29, 38, 50]
```

```
print(list)
```

```
for i in list:
```

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

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

```
print("list after removing even numbers:")
```

```
print(list)
```

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

Date: 27-01-2021

Program No: 17

Aim: Python program to find gcd of number

```
def gcd(a, b):
```

```
    if (b == 0):
```

```
        return a
```

```
    return gcd(b, a % b)
```

```
a = 144
```

```
b = 12
```

```
if (gcd(a, b)):
```

```
    print('GCD of', a, 'and', b, 'is', gcd(a, b))
```

```
else:
```

```
    print('not found')
```

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

Output:

Enter a number: 5

The factorial of 5 is 120

Date: 03-02-2021

Program No: 18

Aim: Python program to find factorial of a number

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

```
factorial=1
```

```
if num<0:
```

```
    print("Sorry, factorial does not exist for negative number.")
```

```
elif num==0:
```

```
    print("The factorial of 0 is 1")
```

```
else:
```

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

```
        factorial = factorial*i
```

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

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

Output:

Enter the number of terms: 5

Fibonacci sequence:

0
1
1
2
3

Date: 03-02-2021

Programs No: 19

Aim: Python program to find fibonacci series of N terms

```
def recur_fibo(n):
```

```
    if n <= 1:
```

```
        return n
```

```
    else
```

```
        return (recur_fibo(n-1) + recur_fibo(n-2))
```

```
nterms = int(input("Enter the number of terms: "))
```

```
if nterms <= 0:
```

```
    print("Please enter a positive integer")
```

```
else:
```

```
    print("Fibonacci sequence: ")
```

```
    for i in range(nterms):
```

```
        print(recur_fibo(i))
```

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

Output:

streaming

streamingly

Date: 03-02-2021

Program No: 20

Aim: Python program to perform string function

```
def add_string(st1):  
    length = len(st1)  
    if length > 1:  
        if st1[-3:] == 'ing':  
            st1 += 'ly'  
        else:  
            st1 += 'ing'  
    return st1  
print(add_string('stream'))  
print(add_string('streaming'))
```

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

Output :

22

Output :-

Enter a number: 4444

Enter a number: 9999

4624

6084

6400

8464

Date : 03/02/2021

Program No: 21

Aim: Python program to perform the sum of given items

```
numbers = [1, 2, 3, 4, 5, 2, 5]
```

```
sums = sum(numbers)
```

```
print (sum)
```

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

Date : 03/02/2021

Program No: 22

Aim: Python program to find perfect even square numbers in a Range

```
num1 = int (input ("Enter a number: "))
```

```
num2 = int (input ("Enter a number: "))
```

```
for i in range (num1, num2+1):
```

```
    for j in range (32, 100+1):
```

```
        if i == j*j:
```

```
            string = str(i)
```

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

```
                print(i)
```

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

Output :-

Enter a number : 4

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

Date: 03/02/2021

Program No: 23

Aim: Python program to display the given pyramid with step number accepted from user.

```
lines = int(input("Enter a number: "))
```

```
i = 1
```

```
j = 1
```

```
while i <= lines:
```

```
    j = 1
```

```
    while j <= i:
```

```
        temp = i * j
```

```
        print(temp, end = " ", flush = True)
```

```
        print(" ", end = " ", flush = True)
```

```
        j = j + 1
```

```
    print(" ")
```

```
    i = i + 1
```

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

Output :-

```
{'h': 2, 'e': 2, 'l': 2, 'o': 3, ' ': 3, 'a': 1, 'y': 1, 'u': 1}
```

Date : 02/02/2021

Program No: 24

Aim: Python program to count the number of characters in a string

```
def char-frequency(str1):
```

```
    dict = {}
```

```
    for n in str1:
```

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

```
        if n in keys:
```

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

```
        else:
```

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

```
    return dict
```

```
print(char-frequency('hello how are you'))
```

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

Output :-

Longest word : morning

length of the longest word : 7

Date : 03/02/2021

Program No: 25

Aim : Python program to accept a list of words and return length of longest words.

```
def find(word):
```

```
    w1 = []
```

```
    for n in word:
```

```
        w1.append(len(n), n))
```

```
    w1.sort()
```

```
    result = w1[-1][0], w1[-1][1]
```

```
    print("Longest word:", result[1])
```

```
    print("Length of the longest word:", result[0])
```

```
find(['hello', 'morning', 'hi'])
```

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

Output:-

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

Date: 03/02/2021

Program No: 26

Aim: Python program to constant Pattern using nested loop

def starc():

n=5

for i in range(n):

for j in range(i):

print("*", end=" ")

print(" ")

for i in range(n, 0, -1):

for j in range(i):

print("*", end=" ")

print(" ")

starc()

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

Output :

The factors of 232 are

1
2
4
8
29
58
116
232

Date : 03/02/2021

Program No: 27

Aim: Python program to print 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)
```

```
print_factors(232)
```

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

Output :-

Enter the length of a side of square

Enter the value : 2

Enter the length and breadth of rectangle

Enter your value : 4

Enter your value : 2

Enter the base and height of triangle

Enter your value : 3

Enter your value : 2

Area of square is 4

Area of rectangle is 8

Area of triangle is 3.0

Date : 03/02/2021

Program No: 28

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

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

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

Output:-

Enter final year : 2040

Leap years:

2024

2028

2032

2036

2040

Date : 17/02/2021

Program No : 29

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

```
import datetime
```

```
a = datetime.datetime.now()
```

```
a = int(a.year)
```

```
b = int(input("Enter final year:"))
```

```
print("\n Leap Years:")
```

```
for i in range(a, b+1):
```

```
    if (i%4==0):
```

```
        print(i)
```

Result:-

The program has been executed and the output was verified.

Output:-

Original list: [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -6]

positive integer list: [1, 2, 9, 87, 24]

Date: 17/02/2021

Program No: 30

Aim: Python program to find positive list of numbers from a given list of integers.

```
list1 = [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -6]
```

```
pos = list()
```

```
for i in list1:
```

```
    if i > 0:
```

```
        pos.append(i)
```

```
print('Original list:', list1)
```

```
print('Positive integer list:', pos)
```

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

Output :-

Enter first number : 5

Enter second number : 6

Enter third number : 8

8 is the biggest number

Date : 17/02/2021

Program No: 31

Aim: Python program to find biggest of 3 numbers entered

```
a = int(input('Enter first number: '))
```

```
b = int(input('Enter second number: '))
```

```
c = int(input('Enter third number: '))
```

```
if a > b and b > c:
```

```
    print(a, 'is the biggest number')
```

```
elif b > a and b > c:
```

```
    print(b, 'is the biggest number')
```

```
else:
```

```
    print(c, 'is the biggest number')
```

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

Output :-

Enter colours separated by commas : red, black, yellow

First color : red

Last color : yellow

Date: 17/02/2021

Programs No: 32

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

```
colours = input('Enter colours separated by commas:').split(',')
```

```
print('First color:', colours[0])
```

```
print('Last color:', colours[-1])
```

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

Output:-

Enter colours separated by commas: red, yellow, brown

Enter colours separated by commas: black, white, green

Colours in color-list1 not contained in colorlist2 are:

['brown', 'red', 'yellow']

Date: 17/02/2021

Program No: 33

Aim: Python program to print all colours from color-list1 not contained in color-list2

```
color1 = set((input('Enter colour separated by commas: ')  
              .split(',')))
```

```
color2 = set((input('Enter colour separated by commas: ')  
              .split(',')))
```

```
print('Colours in colorlist1 not contained in colorlist2  
are: ', list(color1.difference(color2)))
```

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 of 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 48.00 units

Enter the radius of the sphere : 2

Perimeter 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 the length of the cuboid : 4

Enter breadth of the cuboid : 7

Enter height of the cuboid : 2

Date : 17/02/2021

Program No: 34

Aim: Python program to create a package graphics with modules rectangle, circle and sub-package 3-D graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that find area and perimeter of figures by different importing statements.

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' % (2 * 3.14 * r), 'unit')

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')

Total surface area of cuboid with dimensions
40, 70, 20 is 100.00 sq. units

Enter the radius of the sphere: 1

Area of sphere with radius 1.0 is 12.56 sq. units.

sphere.py

def area(r):

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

def perimeter(r):

print('Perimeter 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 perimeter(l, b, h):

print('Perimeter of cuboid with dimensions', l, 'b', 'h', 'is:',
'%.2f' % (4*(l+b+h)), '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:'))

perimeter(a, b)

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

circle.circumference(r)

```

l = float (input ('Enter length of the cuboid:'))
h = float (input ('Enter height of the cuboid:'))
b = float (input ('Enter breadth 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-graphics 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 the length of the cuboid:'))
b = float (input ('Enter the breadth of the cuboid:'))
h = float (input ('Enter the 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 :-

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

Date : 17/02/2021

Program No: 35

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 comp(self, obj):

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

print('Rectangle with length = ' + str(self.length) + ' and breadth = ' + str(self.breadth) + ' has the greater area')

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

print('Rectangle with length = ' + str(obj.length) + ' and breadth = ' + str(obj.breadth) + ' has the greater area')

else:

print('They have equal area')

11. Rectangle (9, 3)

12. Rectangle (3, 4)

11. cmp (12)

Result:- The program has been executed and the 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: 300000

Rs. 300000.0 deposited! current balance is Rs. 400000

Enter amount to withdraw: 5000

Rs. 5000 withdrawn! current balance is: Rs. 395000.0

Date: 17/02/2021

Program No: 36

Ques: Python program to 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.

class BankAccount:

def __init__(self, a, h, t, b):

self.acno = a

self.name = h

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, 'withdraw! current balance is Rs.', self.bal)

else:

print('Insufficient balance to make this transaction!')

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

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


```
t = input('Enter account type:')
```

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

```
ac1 = BankAccount(a, h, t, b)
```

```
ac1.deposit(float(input('Enter amount to deposit:')))
```

```
ac1.withdraw(float(input('Enter amount to withdraw:')))
```

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

Output :-

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

Date: 17/02/2021

Program No: 37

Aim: Python program to create Rectangle class with attributes length and breadth and methods to find area and perimeter, compare 2 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')
```

```
    else:
```

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

r1 = Rectangle (9,3)

r2 = Rectangle (3,4)

u1.cmp(r2)

Result :- The program has been executed and the 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!

Date: 17/02/2021

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.

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!')

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

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

R1 = Rectangle(l, w)

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

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

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

$R_1 < R_2$

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

Output :-

Book title : Programming with Python

Author : GV Rossini

Publisher : ABC Books

Price : 565.9

No. of Pages : 250

Date : 17/02/2021

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 overloading.

class publisher:

def __init__(self, name1):

self.name = name1

def show(self):

pass

class Book (Publisher):

def __init__(self, title1, author1, name1):

self.title = title1

self.author = author1

publisher.__init__(self, name1)

def show(self):

pass

class python (Book):

def __init__(self, p, no, title1, author1, name1):

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(565.90, 250, 'Programming with Python', 'GV Rossam',
'ABC Books')

P1.show()

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. 'n', They are distinguished from dump trucks by configuration: a dumped ']

Date: 21/02/2021

Program No: 40

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

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

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

Output :-

They are distinguished from dump trucks by configuration. a dumper is usually an open 4- wheeled vehicles and the load ship in front of the driver.

Date: 21/02/2021

Programs No: 41

Aim: Python Program to copy odd lines of one file to other

```
a = open('demo.txt', 'r')
b = open('t.txt', 'w')
c = a.readlines()
for i in range(0, len(c)):
    if (i % 2 != 0):
        b.write(c[i])
    else:
        pass
b.close()
b = open('t.txt', 'r')
d = b.read()
print(d)
a.close()
b.close()
```

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

Output :-

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

Date : 21/02/2021

Program No : 42

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

import csv

with open('temp.csv', newline='') as csvfile:

dr = csv.reader(csvfile, delimiter=',', quotechar='"')

for r in dr:

print(' '.join(r))

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

Output

ID Department Name

0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10

Date : 21/2/2021

Program No: 43

Aim: Python program to read specific column of a given csv file and print the content of the column

```
import csv
with open('dep.csv', newline='') as csvfile:
    d = csv.DictReader(csvfile)
    print("ID Department Name")
    print("-----")
    for i in d:
        print(i['value'], i['data'])
```

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

Output:-

- 1, Suzanne Collins, The Hunger Games
- 2, J.K. Rowling, Mary GrandPré, Harry Potter and the Philosopher's Stone
- 3, Stephenie Meyer, Twilight

Date: 21/2/2021

Program No: 44

Aim: Python program to write a python dictionary to a csvfile
After writing the csvfile read the csv file and display the content.

```
import csv
```

```
field_name = ['best-book-id', 'author', 'original-title']
```

```
book = [{'best-book-id': '1', 'author': 'Suzanne Collins', 'original-title':  
'The Hunger Games'}, {'best-book-id': '2', 'author': 'J.K. Rowling,  
Mary GrandPré', 'original-title': 'Harry Potter and the Philosopher  
Stone'}, {'best-book-id': '3', 'author': 'Stephenie Meyer', 'original-title':  
'Twilight'}]
```

```
with open('c1.csv', 'w') as csvfile:
```

```
    writer = csv.DictWriter(csvfile, fieldnames=field_name)
```

```
    writer.writerow()
```

```
    writer.writerow(book)
```

```
with open('c1.csv', newline='') as csvfile:
```

```
    d = csv.reader(csvfile, delimiter=',')
```

```
    for r in d:
```

```
        print(' ', join(r))
```

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