

Output

Enter the value for : 3

Area is 28.2600

Output

Enter the first number : 2

Enter the second number : 4

Enter the third number : 5

The largest number is 5

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 for:"))
print("Area is %.6f" % area(num))
```

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

16/1/2021 Program No 2

AIM: Python program to find largest among 3 numbers.

```
number1 = float(input("Enter the first number:"))
```

```
number2 = float(input("Enter the second number:"))
```

```
number3 = float(input("Enter the third number:"))
```

```
if (number1 > number2) and (number1 > number3):
```

```
    largest = number1
```

```
elif (number2 > number1) and (number2 > number3):
```

```
    largest = number2
```

```
else:
```

```
    largest = number3.
```

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

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

## Output

Enter an integer number: 4  
Square of 4 is 16.

## Output

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

16/01/21 Program No:3

AIM: Python program to find square of a number

```
digit = int(input("Enter an integer number:"))
```

Square = digit \* digit

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

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

26/01/21 Program No:4

AIM: Python program to find area of a circle.

```
from math import pi
```

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

```
print("The area of the circle with radius " + str(r) + " is: " +  
     + str(pi * r ** 2))
```

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

## 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

## Output

Given string:

Hello.. how are you

The vowels present in the string

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

26/01/21 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, square)
```

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

26/1/21 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 out was verified.

## Output

{'when': 1, 'you': 2, 'change': 2, 'the': 2, 'quality': 2,  
'of': 2, 'your': 2, 'thinking': 1, 'life': 1, 'sometimes': 1  
instantly: 1}

26/1/21 Program No: 7

AIM: Python program to count words in a sentence

def word\_count(st):

counts = dict()

word = st.split()

for word in words:

if word in counts:

counts[word] += 1

else:

counts[word] = 1

return counts

print(word\_count("When you change the quality of your thinking, you change the quality of your life sometimes instantly!"))

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

Output

Count of a in the list is: 5

Output

both list have equal length

26/01/21 Program No: 8

AIM: Python program to count a in a list

```
a = ['antony', 'ram', 'rohan', 'salaria']
```

```
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

26/01/21 Program No: 9

AIM: Python program to check the length of list.

```
list1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]
```

```
list2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]
```

```
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 been executed and the output was verified.

Output

both list have equal sum

Output

there are common elements.

26/11/21 Program No: 10

AIM: Python program to check the sum of list

```
list1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]
```

```
list2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]
```

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

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

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

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

```
else:
```

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

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

27/11/21 Program No: 11

AIM: Python program to check the common elements in the list-

```
list1 = [10, 10, 11, 12, 13, 14, 16, 15, 16, 12]
```

```
list2 = [10, 10, 11, 12, 12, 16, 14, 16, 15, 19, 12]
```

```
for value in list1
```

```
    if value in list2:
```

```
        common = 1
```

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

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

```
else:
```

```
    print('no common elements')
```

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

Output

ref \$e\$sh.

Output

aananab.

27/11/21 Program No: 12.

AIM: Python Program to replace a character.

```
def change_char(st1):  
    char = st1[0]  
    st1 = st1.replace(char, '$')  
    st1 = char + st1[1:]  
    print(change_char('refresh'))
```

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

27/11/21 Program No: 13.

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

```
def change_string(st1):  
    return st1[-1] + st1[1:-1] + st1[0]
```

```
print(change_string('banana'))
```

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

Output

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

Output

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

Dictionary in ascending order by value:

[0, 8] [2, 1] [1, 2] [4, 3] [3, 4]

Dictionary in descending order by value:

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

## 27/1/21 Program No:14

AIM: Python program to merge 2 dictionaries.

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

dict1 = {'a': 10, 'b': 8}
dict2 = {'a': 6, 'c': 4}
print(Merge(dict1, dict2))
print(dict2)
```

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

## 27/1/21 Program No:15

AIM: Python program to ascend and descent dictionary

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('original dictionary:', d)
sorted_d = sorted(d.items(), key=operator.itemgetter(1))
print('Dictionary is ascending order by value:', sorted_d)
sorted_d = dict(sorted(d.items(), key=operator.itemgetter(1), reverse=True))
print('Dictionary is descending order by value:', sorted_d)
```

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

Output

[11, 22, 33, 44, 55]

list after removing, [11, 33, 55]

Output

GCD of 98 and 56 is 14

27/1/21 Program No: 16

AIM: Python program to remove even numbers from the list

```
list = [11, 22, 33, 44, 55]
print(list)
for i in list:
    if (i%2 == 0):
        list.remove(i)
print("List after removing: " + str(list))
```

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

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 = 98

b = 56

```
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: 6

The factorial of 6 is 720.

Output

How many terms? 5

0

①

1

2

3

## 3/2/21 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 numbers")
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.

## 3/2/21 Program No: 19

AIM: Python program to find fibonacci sequence.

```

def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return (recur_fibo(n-1) + recur_fibo(n-2))
nterms = int(input("How many terms?"))
if nterms <= 0:
    print("please enter a positive integer")
else:
    print('Fibonacci sequence')

```

## Output.

ceiling  
ceilingly

```
for i in range(terms):
    print(result.fib(i))
```

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

Program No: 20

AIM: Python program to perform string function

```
def add_string(stri)
    length = len(stri)
    if length > 1:
        if stri[-3:] == 'ing':
            stri += 'ly'
        else:
            stri += 'ing'
    return stri
print(add_string('code'))
print(add_string('coding'))
```

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

Output

22

Output

Enter a number: 4444

Enter a number: 9999

4624

60824

6400

8464

## Program No: 21

AIM: Python program to perform the sum of given items

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

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

```
print(sum)
```

Result:

The program has been executed and the output was verified.

## 3/2/2021 Program No: 22

AIM: Python program to find perfect even square number 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.

03/2/21 Program No: 23.

AIM: Python program to Display the given pyramid with step numbers 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:

{'W': 2, 'L': 2, 'U': 2, 'O': 3, 'I': 3, 'W': 1, 'a': 1, 'Y': 1, 'g': 1, 'V': 1}

## 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.

## Program No: 25

AIM: Python program to accept a list of words and return length of largest word.

```
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: " + str(result[0]))
```

Output

largest word: morning

length of the longest word: 7

Output

\*  
\* \*  
\* \* \*  
\* \* \* \*  
\* \* \* \* \*  
\* \* \* \*  
\* \* \*  
\* \*  
\*

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

Result

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

03/06/2011 Program No: 26

AIM: Python program to construct pattern using nested loop.

def star()

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(" ")

star()

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

## Output

Enter the length of a side of square.

Enter your 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: 4

Area of rectangle: 8

Area of triangle: 3.0

### 3) also 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.

### Program No: 28.

AIM: Python program to write lambda function to find area of a square, rectangle, and triangle.

```
print("Enter the length of a side of a square:")
s= int(input("Enter your value:"))
print("Enter the length and breadth of a 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
```

## Output.

Enter final year: 2040

Input years:

2024

2028

2032

2036

2040.

```
print ("Area of a 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.

## Python Program No: 29

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

```
import datetime
a = datetime.datetime.now()
a = int(a.year)
b = int(input("Enter the final year: "))
print("In 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, -5, 4, 8, 7, -3, 3, -7, 6, 2, 4, -6, 7]

Positive integer list : [1, 2, 9, 8, 7, 2, 4].

## Output

Enter 1st no: 5

Enter 2nd no: 6

Enter 3rd no: 8

8 is the largest number.

## 17/2/21 Program No: 30

AIM: Python program to generate positive list of numbers from a given list of integers.

```
list1 = [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -67]
```

```
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.

## 17/2/21 Program No: 31

AIM: Python program to find the biggest 3 numbers entered.

```
a = int(input("Enter the 1st no: "))
```

```
b = int(input("Enter the 2nd no: "))
```

```
c = int(input("Enter the 3rd no: "))
```

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

```
    print(a, "is the largest number")
```

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

```
    print(b, "is the largest number")
```

```
else:
```

```
    print(c, "is the largest number")
```

## Output

Enter colors separated by commas: red, black, yellow  
First color: red  
Last color: yellow.

## Output

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

## Result

The program has been executed and the output was verified.

## Python Program 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: ')
Colours = Colours.split(',')
print('First colour:', Colours[0])
print('Second Last colour:', Colours[len(Colours)-1])
```

## Result

The program has been executed and the output was verified.

## Python Program No: 33

AIM: Python program to print out all colours from colours-list1 not contained in colours-list2.

```
Colours1 = input("Enter colours separated by commas: ")
Colours1 = Colours1.split(',')
Colours2 = input("Enter colours separated by commas: ")
Colours2 = Colours2.split(',')
print('Colours in Colours list1 not contained in Colours list2 are: ', list(set(Colours1) - set(Colours2)))
```

## Output

Enter length of the rectangle: 4

Enter breadth of the rectangle: 3

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

Enter the radius of the circle: 2

Circumference of a circle with radius 2.0 is: 12.56 units

Enter the length of the cuboid: 5

Enter the breadth of the cuboid: 4

Enter the height of the cuboid: 3

Perimeter of cuboid with dimension 5.0, 4.0, 3.0 is 48.0 units

Enter the radius of the sphere: 2

Perimeter of (great circle of) sphere with radius 2.0 is  
12.56 units.

Enter the radius of the sphere: 2

A large sphere with radius 100 is 12560 square units

Perimeter

## Result

The python program has been executed and the output was verified.

## Program No: 34

**AIM:** Python program to create a package graphics with modules rectangle and 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 program to 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 ** 2))
    'sq units')

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

Enter the length of the rectangle: 2

Enter the breadth of the rectangle: 3

Area of rectangle with side 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 cuboid: 4

Enter the breadth of cuboid: 7

Enter the height of cuboid: 2

Total surface area of cuboid with dimension 4.0, 7.0 and 2.0 is 100.00 sq. unit

Enter the radius of the sphere: 1

Area of sphere with radius 1.0 is 12.56 sq. 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').

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').

Final perimeter.py

import circle

from rectangle import \*

from graphics\_3D import cuboid, sphere.

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

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

perimeter(a, b)

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

circle.circumference(r)

l = float(input('Enter length of cuboid:'))

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

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

cuboid.perimeter(l, b, h)

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

## Sphere perimeter(1)

Find area.py

```
import circle
```

```
from rectangle import *
```

```
from graphic_3D_graphics import cuboid, sphere
```

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

```
b = float(input('Enter the 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 greatest area.

17/2/21 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 object 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 greatest area')
    elif self.area() < obj.area():
        print('Rectangle with length =', obj.length, 'and breadth =',
              obj.breadth, 'has the greatest area')
    else:
        print('They have equal area!')

```

r1 = Rectangle(9, 3)  
r2 = Rectangle(3, 4)  
r1.cmp(r2)

Result

The program has been executed and the output was verified

## Output

Enter account number: 0090090909090

Enter name of the account holder: John

Enter account type: savings

Enter your balance: 100000

Enter amount to deposit: 30,000

Rs 300000 deposit! current balance is Rs 400000.0

Enter amount to withdraw: 5000

Rs 5000.0 withdrawn! Current balance is Rs 395000.0.

17/2/21 Program No. 36.

AIM: 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 bank.

class BankAccount:

```
def __init__(self, a, n, t, b):
    self.acno = a
    self.name = n
    self.type = t
    self.balance = b.
```

```
def deposit(self, a):
    self.balance += a
```

```
print('Rs.', a, 'deposited', 'current balance is Rs.', self.balance)
```

```
def withdraw(self, a):
```

```
if self.balance >= a:
    self.balance -= a
```

```
print('Rs.', a, 'withdrawn', 'current balance is Rs.', self.balance)
```

else:

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

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

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

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

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

```
acc = BankAccount(a, n, t, b)
```

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

## Output

Enter length of 1st rectangle : 4

Enter width of 1st rectangle : 6

Enter length of 2nd rectangle : 3

Enter width of 2nd rectangle : 6

Rectangle with length 3 and width 6 has area

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

Result

The program has been executed and the output was verified.

17/2/21 Program No: 37

AIM: Python program to create a class Rectangle with private attribute length and width. Overload '`<`' operator to compare the area of a rectangle.

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 = ' + str(self.\_\_width) + ',

self.\_\_width, ' has the lesser area!')

elif other.area < self.area:

Print('Rectangle with length = ' + str(other.\_\_length) + ',

and width = ' + str(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 = float(input('Enter length of 2nd rectangle'))

Output

Time[hh:mm:ss] 5:11:41

$w = \text{float}(\text{input}('Enter width of another rectangle'))$   
 $R_2 = \text{Rectangle}(l, w)$   
 $R_1 < R_2$ .

## Result

The program has been executed and the output was verified.

17/2/21 Program No: 38

AIM: Python program to create a class time with private attribute hour, minute and second. Overload '+' operator to find sum of 2 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(12, 25, 45)`

`T2 = Time(16, 45, 56)`

`T1 + T2`

## Result

The program has been executed and the output was verified.

## Output

Book title : Programming with python

Author : George Elliot

Published : ABC Books.

Price Rs. 999.99

No. of pages : 400

17/2/21 Program No: 39

AIM: Python program to create a class Published (name). Derive class Book from published with attribute title and author. Derive class Python from Book with attributes price and no. of pages. Write a program that display information about a Python book. Use base class constructor invocation and method overriding.

class Published:

```
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('Price: Rs', self.price)

Print('No. of pages:', self.no\_of\_pages)

p = Python(999.99, 400, 'Programming with python')

## Output

'A trailer is a vehicle designed for carrying bulk material, often on building sites'. They are distinguished from dump trucks by configuration: a dump body.

'George Eliot' 'ABC Books')  
Pl. show()

Resultt:

The program has been executed and the output was verified.

21/21 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 contain 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 truck by configuration: a dumper is usually an open 4-shaft vehicle with the load skip in front of the engine.

## 21/21 Program No: 41

AIM: Python program to copy odd lines of one file to another.

```
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]"

## Output

ID Depart ment Name

0 0

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

## 21/2/21 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:
    d = csv.reader(csvfile, delimiter=',', quotechar='\'')
    for r in d:
        print(r, join(r))
```

## Result

The program has been executed and the output was verified.

## 21/2/21 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 r in d:
        print(r['value'], r['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

21/2/21 Program No. 44

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

```
import csv
```

```
field_names = ["best_book_id", "author", "original_title"]  
book = [ { 'best_book_id': 1, 'authors': 'Suzanne Collins',  
'original_title': 'The Hunger Games' }, { 'best_book_id': 2,  
'authors': 'J.K. Rowling, Mary GrandPré', 'original_title':  
'Harry Potter and the Philosophers Stone' }, { 'best_book_id': 3,  
'authors': 'Stephenie Meyer', 'original_title':  
'Twilight' } ]
```

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

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

```
    writer.writeheader()
```

```
    writer.writerows(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