

## Program - I

Aim:

write a python program to find area of a circle  
 $(A = \pi r^2)$  [using functions]

```
def findArea(r)
```

```
PI = 3.142
```

```
return PI * (r * r)
```

```
# Driver method
```

```
num = float(input('Enter r value :'))
```

```
print("Area is %.6f" % findArea(num))
```

output :

Enter r value : 2

Area is 12.568000

## PROGRAM-II

Aim:

Python program to find largest among three numbers

```
num1 = int(input ("enter the first number:"))
num2 = int(input ("enter the second number:"))
num3 = int(input ("enter the third number:"))
if (num1>num2) and (num1 > num3):
    print ("num1 is largest")
elif (num2 > num1) and (num2 > num3):
    print ("num2 is larger")
else
    print ("num3 is larger")
```

Output :

Enter the first number : 5

Enter the Second number : 3

Enter The second number: 101.

num 3 is larger

Process finished with exit code 0

## PROGRAM - 3

Aim:

Write a program to find square of a number

```
num = float (input ('enter the number'))
```

```
Square = num ** 2
```

```
print ('square : ', Square)
```

Output :

Enter The number

12

Square : 144.0

process finished with exit code 0

## PROGRAM- 4

Aim:

Python program to find square of n numbers in list

```
num = input ('Enter elements of a list separated by space')
print ("\\n")
ul = num.split()
print ('entered numbers are :', ul)
for i in range (len(ul)):
    ul[i] = int (ul[i])
    print ("squares are:")
    for i in range (len(ul)):
        sqr = ul[i] * ul[i]
        print (sqr)
```

Output

Enter elements of a list separated by spaces 2, 5, 6.

Entered numbers are: ['2', '5', '6']

Squares are:

4

25

36

## PROGRAM - 5

Aim:

Python program to find vowels in a string

```
def isvowel(ch)
    return ch in ['A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o', 'u']
def countvowels(str1)
    Count = 0
    for i in range(len(str1)):
        if (isvowel(str1[i])):
            Count = Count + 1
    return Count
str1 = str(input("enter a string :"))
print ("number of vowels in {} is :".format(str1))
countvowels(str1)
```

output :

enter a string : vowels

number of vowels in 'vowels' is 5.

## PROGRAM - 6

Aim: python program to count words in a sentence

```
test_string = "I love Python"  
res = len(test_string.split())  
print("The number of words in a string are :" + str(res))
```

Output

The number of words in string are : 3

## PROGRAM - 7

Aim:

Python program to count a letter 'a' in a list.

```
sent = str(input("enter items to list with spaces:"))
lst = sent.split()
print('list:' , lst)
count = 0
for i in sent:
    if i == 'A' or i == 'a':
        count = count + 1
print("number of letter (a or A) in list is", count)
```

Output:

enter items to list with spaces : a b c ab ac  
list: ['a', 'b', 'c', 'ab', 'ac']

number of letter (a or A) in list is 3

## PROGRAM-8

Aim:-

Python Program to check the length of a list

```
sent = str(input("enter items to list with spaces:"))
list = len(sent.split())
print("length of the list {0} is : ".format(sent.split(), list))
```

Output:

enter items to list with spaces: 1 2 3, 4, 5,

length of the list ['1', '2', '3', '4', '5', '6'] is: 6

## PROGRAM- 9

Aim:

Python program to check the sum of list

```
lst = []
```

```
num = int(input('How many numbers: '))
```

```
for n in range(num):
```

```
    numbers = int(input('Enter number '))
```

```
    lst.append(numbers)
```

```
print("sum of elements in given list is:", sum(lst))
```

Output:

How many numbers: 3

Enter number 4

Enter number 7

Enter number 8

Sum of elements in given list is : 19

## PROGRAM 10

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

```
list1 = input ("enter elements with space").  
ul = list1 split()  
d = {i for i in ul if ul.count(i)>1}  
print ("duplicate elements", d)
```

output:

enter elements with spaces: 2 3 4 5 2  
duplicate elements { '2' }.

## PROGRAM 11

Aim:

Python program to replace a character string.

```
String = input ("Enter string:")
String = String.replace ('M', 'N')
print ("modified string:")
print (String)
```

Output:

Enter String : Nico

Modified string:

Nice

## Program - 12

Aim:

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

```
string = input ("enter a string")
first = string [0]
last = string [-1]
n = last + string [1:-1] + first
print ("new string is", n)
```

Output :

enter a string : python  
new string : nython

## PROGRAM: 13

Aim:

Python program to merge two dictionaries

```
d1 = {'A': 1, 'B': 2}
```

```
d2 = {'C': 3}
```

```
d1.update(d2)
```

```
print ("concatenated dictionary is :")
```

```
print (d1)
```

Output:

concatenated dictionary is:

{'A': 1, 'B': 2, 'C': 3}

## PROGRAM: 15

Aim:

Python program to remove even numbers from a list.

lis = [1, 2, 3, 4, 5]

out = []

for num in lis:

if num % 2 != 0

out.append(num)

print('list of odd numbers:', out)

Output

list of odd numbers : [1, 3, 5]

## Program 16

Aim:

python program to find gcd of a number

```
def compute_hcf(x,y)
    if x>y
        smaller = y
    else
        smaller = x
    for i in range(1, smaller+1)
        if ((x % i == 0) and (y % i == 0))
            hcf = i
    return hcf
num1 = int(input('enter first number: '))
num2 = int(input('enter second number: '))
print("The H.C.F is", compute_hcf(num1, num2))
```

Output:

Enter the first number 54

Enter the second number 24

The H.C.F. is 6

## Program - 17

Aim:

Python program to find factorial of a number

```
x = int(input("enter a number"))
fact = 1
for i in range(1, x+1):
    fact = fact * i
print("factorial of {} is {}".format(x), fact)
```

Output :

enter a number : 5  
factorial of 5 is 120

## PROGRAM-18

Aim:

Python program to find fibonacci Series

```
x = int(input("enter a limit :"))
```

```
c = 0
```

```
b = 1
```

```
a = 0
```

```
print ("fibonacci Series :")
```

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

```
    print (c)
```

```
    c = a+b
```

```
    a = b
```

```
    b = c
```

```
c = a
```

Result is this ↴

output

enter a limit : 8

fibonacci series

0

1

1

2

3

5

8

13

## Program - 19

Aim:

Python program to perform string functions

```
String 1 = "Apple"  
print ("Initial string:", string 1)  
print ("\n first character of string:", string 1[0])  
print ("\n last character of string:", string 1[-1])
```

Output:

Initial String : Apple

First character of string : A

Last character of string : e

## Program - 20

Aim:

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

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

output :

4624

6084

6400

8466

Process finished with exit code 0

## Program - 21

Aim:

Display the given pyramid with step number accepted from user

Eg:  $\star n = 4$

1  
2 4

3 6 9

4 8 12 16

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

for i in range(1, rows):

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

print(i+j, end=' ')

print()

output

Enter the number of rows: 6

1

2 4

3 6 9

4 8 12 16

## Program - 22

Aim:

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

```
String = input()
if len(string) < 3:
    print(string)
elif string [-3] == 'ing':
    print(string + 'ly')
else:
    print(string + 'ing')
```

andput :

String

Stringly

### Program - 23

Aim:

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

a = []

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

for x in range(0, n):

    element = input("Enter element " + str(x+1) + ":")

    a.append(element)

    max\_l = len(a[0])

    temp = a[0]

    for i in a:

        if (len(i) > max\_l):

            max\_l = len(i)

            temp = i

    print("The word with the longest length is :")

    print(temp)

output:

Enter the number of elements in list : 3

Enter element 1 : banglore ?

Enter element 2 : mysore

Enter element 3 : coorg

The word with the longest length is :

banglore

process finished with exit code 0

## Program - 24

Aim:

Construct following pattern using nested loop

```
rows = int(input("Enter the number of rows:"))
for i in range(0, rows):
    for j in range(0, i+1):
        print("*", end="")
        print(" ")
    for i in range(rows+1, 0, -1):
        for j in range(0, i-1):
            print("*", end="")
            print(" ")
```

Output:

Enter the number of rows 4

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

## program d5

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

r-area = lambda len, ht : len \* ht

t-area = lambda b, ht : b \* ht / 2

c-area = lambda rad : math.pi \* rad \* rad

print ("area of rectangle (30, 20) is", r-area(30, 20))

print ("area of circle (15) is", c-area(15))

print ("area of triangle (12, 20) is", t-area(12, 20))

Output :

area of rectangle (30, 40) is 1200

area of circle (15) is 1256.6370614359173

area of triangle (12, 20) is 120.0

## program 26

work with built-in packages.

```
import platform  
x = platform.system()  
print(x)  
print()  
x = dir(platform)  
print(x)  
print()  
import datetime  
x = datetime.datetime.now()  
print(x)  
x = datetime.datetime.now()  
print(x.year)  
print(x.strftime("%A"))
```

output

2021-06-29 0:8:20: 25.202581

2021

Tuesday

process finished with exit code 0

## Program #7

Create a package graphics with modules rectangle, circle and sub package 3dgraphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements.

### Circle.py

```
from math import pi
def area(r):
    return (pi * r * r)
def perimeter(r):
    return (2 * pi * r)
```

### Rectangle.py

```
def area(l,b):
    return (l * b)
def perimeter(l,b):
    return (2 * (l + b))
```

Inside the folder of graphics.py create another folder named 3dgraphics.py

### Sphere.py

```
from math import pi
def surface_area(r):
    return (4 * pi * r * r)
def volume(r):
    return ((4/3) * pi * r * r * r)
```

### Cuboid.py

```
def surface_area(l,w,h):
    return (2 * l * w) + (2 * l * h) + (2 * w * h)
def volume(l,w,h):
```

```
return (4 * (l + w + h))
```

Outside the folder

```
from graphics.rectangle import *
```

```
from graphics.circle import *
```

```
from graphics.sphere import *
```

```
from graphics.cuboid import *
```

```
print ("Rectangle")
```

```
l = int(input ("enter length :"))
```

```
b = int(input ("enter breadth :"))
```

```
print ("area of rect", vararea(l,b))
```

```
print ("perimeter of rect", rperimeter(l,b))
```

```
print ("circle")
```

```
r = int(input ("enter a radius"))
```

```
print ("area of circle", careal(r))
```

```
print ("peri of circle", cperimeter(r))
```

```
print ("sphere")
```

```
r = int(input ("enter a radius"))
```

```
print ("area of sphere", sareal(r))
```

```
print ("peri of sphere", sperrimeter(r))
```

```
print ("cuboid")
```

```
l = int(input ("enter a len :"))
```

```
w = int(input ("enter a width :"))
```

```
h = int(input ("enter a height :"))
```

```
print ("area of cuboid", cuarea(l,w,h))
```

```
print ("peri of cuboid", cuperimeter(l,w,h))
```

Output:

Rectangle:

Enter length : 5

Enter Breadth : 10

Area of rectangle : 50

Perimeter of rectangle : 30

Circle:

Enter radius of circle : 4

Area of circle : 50.26548295

Perimeter of circle : 25.132741228

Sphere

Enter radius of sphere : 3

Area of sphere : 113.097335529123285

Perimeter of sphere : 113.09733352923283

Cuboid:

Enter length : 7

Enter width : 10

Enter height : 8

Area of Cuboid : 472

Perimeter of cuboid : 108

## program 28

Aim:

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, length, breadth):

self.length = length

self.breadth = breadth

def area(self):

return self.length \* self.breadth

def peri(self):

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

a = int(input("Enter length of rectangle :"))

b = int(input("Enter breadth of rectangle :"))

c = int(input("Enter length of rectangle :"))

d = int(input("Enter breadth of rectangle :"))

obj = Rectangle(a, b)

obj1 = Rectangle(c, d)

print("Area of 1<sup>st</sup> rectangle : ", obj.area())

print("Area of 2<sup>nd</sup> rectangle : ", obj1.area())

print("Perimeter of 1<sup>st</sup> rectangle is : ", obj.peri())

print("Perimeter of 2<sup>nd</sup> rectangle : ", obj1.peri())

if obj.area() == obj1.area():

print("equal")

else:

print("not equal")

Output:

Enter the length of rectangle : 4

Enter the breadth of rectangle : 5

Enter the length of rectangle : 3

Enter the breadth of rectangle : 2

Area of 1st rectangle : 20

Area of 2nd rectangle : 6

Perimeter of 1st

perimeter of 2nd rectangle : 18

rectangle : 10

not equal.

## Program - 29

Aim:

Create a Bank account with members account number, name, type of account and balance with constructor and methods to deposit at the bank and withdraw an amount from the bank.

python code

class Bank Account :

def init\_(self):

self .balance = 0

def deposit\_(self):

amount = float(input ("Enter amount to be deposited"))

self .balance += amount

print ("\\n Amount deposited: ", amount)

def withdraw\_(self):

amount = float(input ("Enter amount to be withdrawn:"))

self .balance -= amount

self .balance = amount

print ("\\n You withdrew: ", amount)

else :

print ("\\n insufficient balance")

def display\_(self):

print ("\\n Net available Balance : ", self .balance)

s = Bank\_Account()

s .deposit ()

s .withdrawal()

s .display()

output:

Balancer amount to be deposited: 900

Amount deposited: 900.0

Balancer amount to be withdrawn: 400.0

Yo withdrawal: 400.0

new available balance: 500.0

## Program - 30

Aim:

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

class A:

-- length = 0

-- width = 0

-- area = 0

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

self.\_\_length = l

self.\_\_width = w

def area(self):

self.\_\_area = self.\_\_length \* self.\_\_width

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

if (self.\_\_area) > (other.\_\_area):

return true

else

return false

rect 1 = A(3, 4)

rect 1.area()

rect 2 = A(6, 5)

rect 2.area()

if (rect1 > rect2):

print("rect1 is greater than rect2")

else:

print("rect2 is greater than rect1")

output:

recd 2 is greater than recd 1;

### Program - 3 |

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('facebook.com'))

Output:

{'t': 1, 'a': 2, 'c': 1, 'b': 1, 'o': 3, 'k': 1, 'l': 1, 'm': 1}

## Program - 32

Aim: Generate all factors of a number.

def print\_factor(x):

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

    for i in range (1, x+1):

        if x % i == 0:

            print (i)

num = 10

print\_factors(num)

Ques:

The factors of 10 are:-

1

2

5

10

# Program 33

Aim:

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

Class Time:

def \_\_init\_\_(self, hour, minute, second):

    self.hour = hour

    self.minute = minute

    self.second = second

def addTime(self, other):

    h = self.hour + other.hour

    m = self.minute + other.minute

    s = self.second + other.second

    self.checkTime(h, m, s)

def checkTime(self, h, m, s):

    s = s

    h = h

    m = m

    if s == 60:

        m = m + 1

    if s > 60:

        m = m + 1

        s = s - 60

    if m == 60:

        h = h + 1

    if m > 60:

        h = h + 1

    m = m - 60

```
print('NEW TIME')
print('{:02}:{:02}:{:02}'.format(h,m,s))
return 0
time1 = input('Enter first time in the format HH:MM:SS/n')
time2 = input('Enter second time in the format HH:MM:SS/n')
h1, m1, s1 = map(int, time1.split(':'))
h2, m2, s2 = map(int, time2.split(':'))
t1 = Time(h1, m1, s1)
t2 = Time(h2, m2, s2)
t1+t2
```

Output :

Enter first time in the format HH:MM:SS

2:2:3

Enter second in the format HH:MM:SS

4:58:59

New TIME

7:1:2

## Program 34:

Aim:

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.

class Publisher:

def \_\_init\_\_(self, pubname):

self.pubname = pubname

def display(self):

print("Publisher Name:", self.pubname)

class Book(Publisher):

def \_\_init\_\_(self, pubname, title, author):

publisher.\_\_init\_\_(self, pubname)

self.title = title

self.author = author

def display(self):

print("Title:", self.title)

print("Author:", self.author)

class python(Book):

def \_\_init\_\_(self, pubname, title, author, price, no\_of\_pages):

Book.\_\_init\_\_(self, pubname, title, author)

self.price = price

self.no\_of\_pages = no\_of\_pages

def display(self):

print("Title:", self.title)

print("Author:", self.author)

print("Price:", self.price)

print('Number of pages:', self.no\_of\_pages)

s1 = Python("ak book", "Taming Python By Programming",

'Iceva Josc', 200, 20)

s1.display()

Output:

Title: Taming python By Programming

Author: Teeva Jose

Price: 200

Number of pages: 219

## program 35

Aim:

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

```
def file_read(fname):  
    with open(fname) as f:  
        file_list = f.readlines()  
    print(file_list)
```

```
file_read('fileread.txt')
```

output:

They are distinguished from dump truck by configuration: a dumper is really an open truck vehicle with the load skip in front of the front.

## Program 36

Aim:

python program do copy odd lines of one file to other

```
fn = open('text1.txt', 'r')
```

```
fn1 = open('text2.txt', 'w')
```

```
cont = fn.readlines()
```

```
cont = tuple(cont)
```

```
for i in range(0, len(cont)):
```

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

```
        fn1.write(cont[i])
```

```
    else:
```

```
        pass
```

```
fn1.close()
```

```
fn1 = open('text12.txt', 'r')
```

```
cont1 = fn1.read()
```

```
print(cont1)
```

```
fn1.close()
```

```
fn1.close()
```

output:

file 1

['python\n', 'c++\n', '(n', 'c#(n', 'php\n', 'html\n']

file 2

c++

(#

html

## Program 37

Aim:

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

```
import csv  
with open ('dep.csv', newline = '') as csvfile:  
    data = csv.reader(csvfile, delimiter = ',', quotechar = "'")  
    for row in data:  
        print (',', join(row))
```

## Program 38

Aim:

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

```
import csv  
with open('dep.csv', newline='') as csvfile:  
    data = csv.DictReader(csvfile)  
    print("CSV")  
    print("-----")  
    for row in data:  
        print(row['MsCode'], row['HFCode'],  
              row['Estimate'])
```

## Program 39

Aim:

Write a python program to write a python dictionary to a csv file. After writing the CSV file read the csv file and display the contents.

```
import csv
```

```
csv_columns = ['id', 'column1', 'column2', 'column3', 'column4',  
               'column5']
```

```
dict_data = {'id': [1, 2, 3],  
            'column1': [33, 25, 56],  
            'column2': [35, 30, 30],  
            'column3': [21, 40, 55],  
            'column4': [71, 25, 55],  
            'column5': [10, 10, 40]}
```

```
csv_file = "dep.csv"
```

```
try:
```

```
    with open(csv_file, 'w') as csvfile:  
        writer = csv.DictWriter(csvfile, fieldnames=csv_columns)  
        writer.writeheader()  
        for data in dict_data:  
            writer.writerow(dict_data)
```

```
except IOError:
```

```
    print("I/O error")
```

```
data = csv.DictReader(open(csv_file))
```

```
print("CSV file as a dictionary:\n")
```

```
for row in data:
```

```
    print(row)
```

output:

"[1, 2, 3]", "[33, 25, 56]", "[35, 30, 30]"  
"[21, 40, 50]", "[71, 25, 55]", "[10, 10, 40]"  
"[1, 2, 3]", "[33, 25, 56]", "[35, 30, 30]",  
"[21, 40, 55]", "[71, 25, 55]", "[10, 10, 40]"