Date : 18-02-2021

PROGRAM 1.

AIM :

Python Program to find area of a circle. ($A = \pi r^2$) [using function]

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

OUTPUT :

Enter the radius : 3

Area of the circle is 28.260000

Date: 18-02-2021

PROGRAM 2

AIM:

Python program to find largest among three numbers.

```
a = int(input("Enter the First Value:"))
b = int(input("Enter the Second Value:"))
c = int(input("Enter the third Value:"))

if (a>=b) and (a>=c):
    largest = a
elif (b>=a) and (b>=c):
    largest = b.
else:
    largest = c
print("The largest number is", largest)
```

OUTPUT:

Enter the first Value : 32.
Enter the Second Value : 26.
Enter the Third Value : 55
The largest number is 55

Date: 18-02-2021

PROGRAM 3.

AIM:

Python program to find square of a number

```
num = int(input("Enter the Number: "))
square = num * num;
print("The square of {0} is {1}".format(num,
                                        square))
```

OUTPUT:

Enter the Number : 6.

The Square of 6 is 36.

Date : 21-02-2021

PROGRAM 4.

AIM :

Python program to find square of n numbers in a list.

```
list = [12, 13, 14, 15]
for n in list:
    square = n*n
    print("The square of", n, "is", square)
```

OUTPUT :

-The Square of 12 is 144
-The Square of 13 is 169
-The Square of 14 is 196
-The Square of 15 is 225

Date : 21-02-2021

PROGRAM 5

AIM :

Python program to find vowels in a string

```
def vowel_count(str):
    count = 0
    vowel = set("AaEeIiOoUu")
    for alphabet in str:
        if alphabet in vowel:
            count = count+1
    print("\n Given string :",str)
    print("\n No of vowels in the string :",count)

    res = set([each for each in str if each
                in vowel])
    print("\n The Vowels present in the string:",
                res)
str = "Simple Program"
vowel_count(str)
```

OUTPUT:

The Given String : Simple Program.

No. of Vowels in the String : 4

The Vowels present in the String : {'a', 'o', 'e', 'i'}

Date : 21-02-0001

PROGRAM 6.

AIM:

Python program to count words in a sentence.

```
test_string = "This is a Simple Program"
print("\n The Original String :" + test_string)
res = len(test_string.split())
print("\n The Number of words in the String is"
                                    + str(res))
```

OUTPUT :

The Original String : this is a Simple Program
The Number of words in the String is 5

Date: 23-02-2021

PROGRAM 7

AIM:

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

```
test_str = "Sample Program"

count = 0

for i in test_str:
    if i == 'a':
        count = count + 1

print("\n Count of 'a' in the list = "
      +str(count))
```

OUTPUT :

Count of 'a' in the list = 2

Date : 22-02-2021

PROGRAM 8

AIM :

Python program to check the length of a list

```
test_list = [1, 4, 5, 7, 8, 9]

print("\n List = " + str(test_list))

counter = 0
for i in test_list:
    counter = counter + 1

print("\n Length of list is" + str(counter))
```

OUTPUT.

list = [1,4,5,7,8,9]

length of list is 6

Date : 22-02-2021

PROGRAM 9

AIM :

Python program to check the sum of list.

```
list = [10, 20, 25, 15, 30]

total = sum(list)

print ("\n list = ", list)
print ("\n Sum of all elements in the given
        list is ", total).
```

OUTPUT:

list = [10, 20, 25, 15, 30]

Sum of all elements in the given list is 100

Date : 23-02-2021

PROGRAM 10

AIM :

Python program to check the common element ent the list

```
def common_member (a, b):
    a_set = set (a)
    b_set = set (b)

    if (a_set & b_set):
        print ("\nCommon elements :", a_set & b_set)
    else:
        print ("\n No Common elements")

a = [1, 2, 3, 4, 5]
b = [4, 5, 3, 8, 9]
common_member (a, b)
```

OUTPUT:

Common Elements : {3, 4, 5}.

Date: 23-02-2021

PROGRAM 11

AIM :

Python program to replace a character in a string

```
str1 = input("Enter the String:")
ch = input("Enter the character you want
            to Replace:")
newch = input("Enter the New character:")

str2 = str1.replace(ch, newch)

print("\n Original String :", str1)
print("\n New Modified String :", str2).
```

OUTPUT :

Enter the string : sample

Enter the character You want to Replace: a

Enter the New character : i

Original string : sample

New Modified string : simple

Date : 23-02-2021

PROGRAM 12.

AIM :

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

```
str = input ("Enter a string :")
new_str = str [-1:] + str [1:-1] + str [:1]
print (new_str).
```

OUTPUT.

Enter a string : program.

mrograp.

Date : 27-02-2021.

PROGRAM 13.

AIM :

Python program to merge two dictionaries

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

dict1 = {'a': 10, 'b': 8}.
dict2 = {'d': 6, 'c': 4}.

print (Merge (dict1, dict2))
print (dict2)
```

OUTPUT :

None

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

Date: 28-08-2021.

PROGRAM 14

AIM:

Python program to ascend and decent dictionary.

```
import operator.
d = {20: "Kavin", 4: "Bibin", 3: "Anes", 1: "Rose",
     17: "Shivani"}.
print("In Original dictionary:\n", d).
sorted_d = sorted(d.items(), key = operator.
                  itemgetter(1))
print("In Dictionary in ascending order:\n", sorted_d)
sorted_d = dict(sorted(d.items(), key = operator.
                itemgetter(1), reverse = True))
print("In Dictionary in descending order:\n",
      sorted_d)
```

OUTPUT :

Original dictionary :

{20: 'kevin', 4: 'bibin', 3: 'arei', 1: 'rose', 17: 'shivam'}

Dictionary in ascending order :

{(3, 'arei'), (4, 'bibin'), (20, 'kevin'), (1, 'rose'), 17, 'shivam'}

Dictionary in descending order :

{{17, 'shivam', 1: 'rose', 20: 'kevin', 4: 'bibin', 3: 'arei'}

Date : 28-02-2021

PROGRAM 15

AIM :

Python program to remove even numbers from a list

```
list = [11, 22, 33, 44, 55, 68]

print("\n Original list :", list)

for i in list:
    if (i%2 == 0):
        list.remove(i)

print("\n list after removing even numbers",
      list)
```

OUTPUT:

Original list: [11, 22, 33, 44, 55, 68]

list after removing even numbers: [11, 33, 55]

Date: 01-08-2021

PROGRAM 16.

AIM:

Python program to find gcd of a number.

```
def gcd(a,b):
    if (a==0):
        return b
    if (b==0):
        return a
    if (a==b):
        return a
    if (a>b):
        return gcd(a-b, b)
    return gcd(a, b-a)

a=98
b=56
if (gcd(a,b)):
    print("\n GCD of", a "and", b, "is", gcd(a,b))
else:
    print("not found")
```

OUTPUT

GCD of 98 and 56 is 14.

Date : 01-03-2021

PROGRAM 17.

AIM:

Python program to find factorial of a number.

```
num = int(input("Enter the element:"))

factorial = 1

if num < 0:
    print("\n factorial does not exist for
    negative numbers")
elif num == 0:
    print("\n the factorial of {0} is {1}")
else:
    for i in range(1, num + 1):
        factorial = factorial * i
    print("\nthe factorial of", num, "is", factorial)
```

OUTPUT.

Enter the Element : 5

The factorial of 5 is 120

Date: 01-03-2021

PROGRAM 18.

AIM:

Python program to find fibonacci series

```
def recur_fibo(n):
    if n <= 1
        return n
    else:
        return (recur_fibo(n-1) + recur_fibo(n-2))

nterms = int(input("enter the limit:"))

if nterms <= 0:
    print("enter a positive integer")
else:
    print("fibonacci sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

OUTPUT :

Enter the limit : 8.

Fibonacci sequence.

0
1
1
2
3
5
8
13

Date : 01-03-2021

PROGRAM 19.

AIM :

Python program to perform string functions.

```
string1 = "Simple"
print("Initial string :", string1)
print("\n First character of string :", string1[0])
print("\n Last character of string :", string1[-1])
```

OUTPUT :

Initial string : Simple

First character of string : s

Last character of string : e

28/3/21

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, 10000, 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

8464

28-3-21

PROGRAM 21

AIM:

Display the given pyramid with step number accepted from user.

Eg: 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 : 4.

1

2 4

3 6 9

4 8 12 16.

28-3-21

PROGRAM 22

AIM:

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

{'g': 1, 'a': 1, 'c': 2, 'e': 1, 'b': 1, 'd': 3, 'k': 1
'i': 1, 'm': 1}

28-3-21

PROGRAM 23.

AIM:

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

```
def add_string(str1):
    length = len(str1)

    if length > 2:
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else:
            str1 += 'ing'
    return str1
print(add_string('crying'))
print(add_string('thistle'))
print(add_string('string'))
```

DUYPOY.

Cayingly
thinking
Stengly.

28-3-21

PROGRAM 24

AIM:

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

```
def RongestLength(a):
    max1 = len(a[0])
    temp = a[0]
    for i in a:
        if(len(i)>max1):
            max1 = len(i)
            temp = i
    print("The word with the longest length is:", temp, "and length is", max1)
a = input("Enter a list elements separated by spaces")
a = a.split()
LongestLength(a)
```

OUTPUT:

Enter a list elements seperated by spaces
India is my country
The word with the longest length is:
country and length is 7.

28-3-21

PROGRAM 25

AIM:

Construct following pattern using nested loops.

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

```
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: 5

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

PROGRAM 26.

AIM:
Generate all factors of a number.

```
def print_factors(x):
    print("the factors of", x, "are:")
    for i in range(1, x+1):
        if x % i == 0:
            print(i)

num = 10
print_factors(num)
```

OUTPUT:

The factors of 10 are

1

2

5

10

PROGRAM 27.

AIM:

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

```
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(80,20) is", r_area(10,20))
print("area of circle(15) is:", c_area(20))
print("area of triangle(12,20) is", t_area(22,20))
```

OUTPUT

area of rectangle (30,20) is 200

area of circle (15) is : 1256.6370614359172

area of triangle (12,20) is 220.0.

PROGRAM 28.

AIM:
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

2021-03-28 04:18:33.347291

2021

Sunday

PROGRAM 29.

AIM:

Create a package graphics with modules rectangle, circle and subpackage 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each method. Write programs that finds area and perimeter of figures by different importing statements (Include selective import of modules & import * statements.

Create a folder graphics. py.

circle.py.

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

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 dgraphics.py.

Sphere.py.

```
from math import pi
def area(r):
    return (4*pi*r*r)
def perimeter(r):
    return ((4/3)*pi*r*r*r)
```

Cuboid.py.

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

```
def cuperimetre (l, w, h):
return (4 * (l+w+h))
```

Outside the folder.

```
#//area.py.
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", rearea (l,b))
print ("Perimeter of rect! rperimetre(l,b))
print ("Circle")
r = int (input ("Enter a radius").
print ("Area of circle:", carea(r))
print ("Perimetre of circle", cperimetre (r))
print ("Sphere")
```

```
r = int(input("Enter a radius"))
print("Area of sphere", sarea(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", carea(l, w, h))
print("Perimeter of cuboid", cperimeter(l, w, h)).
```

OOPS04

Rectangle
Enter length : 5
Enter breadth : 10
Area of rectangle : 50
Perimeter of rectangle : 30

Circle.
Enter radius of circle : 4
Area of circle : 50.265482045
Perimeter of circle : 25.132741228

Sphere.
Enter radius of sphere : 3
Area of sphere : 13.097335529233453
Parmeter of sphere : 113.097335529233453

Cuboid.
Enter length : 7.
Enter width : 12.
Enter height : 8
Area of cuboid : 472.
Perimeter of cuboid : 108

PROGRAM - 30

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 1st rectangle:", obj.area())
print("Area of 2nd rectangle:", obj1.area())
print("Perimeter of 1st rectangle is:", obj.peri())
print("Perimeter of 2nd rectangle is:", 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 rectangle : 18

Perimeter of 2nd rectangle : 10.

Not Equal

PROGRAM - 31.

AIM: Create a Bank account with members account and balance. Write constructor and methods to deposit at the Bank and withdraw an amount from the Bank.

```
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:"))
    if self.balance >= amount:
        self.balance -= amount
        print("\n You withdrawn:", amount)
```

```
else:
    print ("\n Insufficient Balance")
def display (self):
    print ("\n Net Available Balance =", self.balance).
s = Bank.Account ()
s.deposit ().
s.withdraw.
s.display.
```

OOPD-1.

Enter amount to be deposited :900

Amount Deposited : 900.0

Enter amount to be withdrawn : 400

You withdraw : 400.0

Net available balance : 500.0 .

PROGRAM 32

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 __lt__(self, other):
        if (self.__area > other.__area):
            return true
        else:
            return false
```

```
rect1 = A(3, 4)
rect1.area()
rect2 = A(6,5)
rect2.area()
if (rect1 > rect2):
    print ("rect1 is greater than rect2")
else:
    print ("rect2 is greater than rect1")
```

OUTPUT :

rect 2 is greater than rect 1.

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 __add__(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 ('{0}:{1}:{2}'.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 form 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).

OUTPUT

Enter first time is the format HH:MM:SS.

2:2:3.

Enter second time is 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 of pages. Write a program that displays information about a Python Book. Use base class constructor invocation & 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_page= no_of_page.
  def display(self):
     print ("title:" self; title)
     print ("Author:", self.author)
     print ("Price:" , self.price)
     print ("Number of page :", self.no_of_page)
b1= Python ("abc books", "learning Python By
  Programming", "Jeeva Jose', 200, 219)
b1.display()
```

ᐅᑐᐊᑎᑐᐊ -

Title : Learning Python By Programming

Author : Jeeva 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('demo.txt').
```

OUTPUT

'Python is a general-purpose code language'. which means that, unlike HTML, CSS, Javascript it can be used for other type types of programming and soft development besides web development

PROGRAM 36.

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

```
f = open("demo.txt", "r")
f1 = open("demo4.txt", "w")
c = f.readlines()
for i in range(0, len(c)):
    if (i%2 1 = 0):
        f1.write(c[i])

    else:
        pass.

f1.close
f1 = open("demo4.txt", "r")
c = f1.read()
print(c).

f.close()
f1.close()
```

OUTPUT.

It can be used for other type of programming and software development besides web development.

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))
```

Output.

SN , Movie , Rating

1 , Lord of the rings , 5

2 , Harry potter , 6 .

PROGRAM - 38

AIM:
Write a python program to read specific columns of a given csv file & 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['1-IECCode'],
row['Estimate'])
```

PROGRAM - 39

AIM:

Write a Python program to write a Python dictionary to a csv file. After writing the csv file read & display the content

```
import csv.
csv_columns = ['id', 'Column1', 'Column2',
               'Column3', 'Column4', 'Column5']
dict_data = {'id': ['1','2','3']
'Column1': [33,25,56],
'Column2': [85,30,80];
'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

{'id': "['1', '2', '3']", 'column1': '[33,25,56]', 'column2':

{'id': ['1', '2', '3'], 'column1': '[33,25,56]', 'column2':

{'id': "['1', '2', '3']", 'column1': '[33,25,56]', 'column2':

{'id': "['1', '2', '3']", 'column1': '[33,25,56]', 'column2':

{'id': "['1', '2', '3']", 'column1': '[33,25,56]', 'column2':

{'id': ['1', '2', '3']", 'column1': '[33,25,56]', 'column2':