Phython Lab Record.

Program No: 1 Date: 18-02-2021

Aim: To find area of a circle

Program:-

```
import math

def area_of_circle(r):
    a = r**2 * math.pi
    return a

r = float(input("enter the radious of circle:"))
Print("%.2f" % area_of_circle(r))
```

~~Outpu~~

Result: Program executed and Output verifyed

Result

enter the radius of circle: 2

12.566371

enter the radius of circle:

28.274334.

Program No: 2 Date: 18-02-2021

Aim: To find largest among three numbers

Program:

```
a = int(input("enter the first number:"))
b = int(input("enter the second number:"))
c = int(input("enter the third number:"))
if (a>b) and (a>c):
    largest = a
elif(b>a) and (b>c):
    largest = b
else:
    largest = c
    Print("the largest is:", largest)
```

Result: Program executed and output verifyed

Result

enter the first number : 5

enter the Second number : 4

enter the third number : 9

the largest is : 9

Program NO: 3 Date: 18-02-2021

Aim: To find Square of a number

Program

```
a = int(input("enter the number:"))
Square = a * a
Print("Square of {0} is {1}".format(a, Square))
```

Result: Program has been executed output verifyed

Result

Enter the number : 5

Square of 5 is 25

Program NO: 4 Date: 21-02-2024

Aim: Python Program to find Square of n numbers in a list

[list = [12, 13, 14, 15]

for n in list:

Print(n*n)]

Program:

```
a = [11, 12, 13, 14]
SquaredValues = [number * number
                 for number in a]

Print ("Original Values :", a)
Print ("Squared Values:", SquaredValues)
```

Result : Program is executed and output verifyed

Result

Original Values: [11, 12, 13, 14]

Squared Values: [121, 144, 169, 196]

Date: 21-02-2021

Program: 5

Aim: Python Program to Find vowels in a String

Program:

```
String = input("enter the string:")
vowels = 'AaEeIiOoUu'
res = set(each for each in String
          if each in vowels)
Print("Vowels Present in the String:",res)
```

Result: Program is executed and output is verifyed

Result:

Enter the String : AmalJyothi

Vowels Present in the String : {'a', 'o', 'i', 'A'}

Date: 21-02-2021

Programe : 6

Aim : Python Program to Count words in a Sentence.

Program:

```
String= input("Enter the String :")
Print ("String is :", String)
    res = len(String.Split())
    Print ("Count of Word is :", res)
```

Result: Program is executed and Output is verifyed

Result:

Enter the String : AmalJyothi College of Engineerin
Kanjirappally.

String is : AmalJyothi College of Engineering
Kanjirappally.

Count of word is: 5

Date: 27-02-2021

Program: 7

Aim: Python Program to Count a letter 'a' in a list.

Program:

```
String = input("enter the String :")
Print ("the String is :", String)
Count = 0
for i in String:
    if i == "a":
        Count = Count + 1
Print ("Count of a is :", Count)
```

Result: Program is executed and output is verifyed

Result.

Enter the String: AmalJyothi

the String is : AmalJyothi

Count of a is : 1

Program : 8

Date : 27-02-2021

Aim: Python Program to Check the length of a list

Program:

```
list1 = [1, 2, 3, 4, 5, 6, 7]
len1 = len(list1)
Print("length of list1 is", len1)
```

Result: Program is executed and output is verified

Result:

length of list1 is 7

Program : 9 Date: 27-02-2021

Aim: Python Program to check the Sum of list.

Program:

```
list1 = [1, 2, 34, 56, 78, 89, 76, 23]
total = sum(list1)
Print("Sum of list1 is:", total)
```

Result: Program is ~~verifyed~~ and output Executed is verifyed

Result

Sum of list1 is : 359

Program 10:

Date: 27-02-2021

Aim: Python Program to check the common element in the list

Program:

```
list1 = Input("enter the element with spaces:")
U = list1.Split()
di = {i for i in U if U.Count(i) > 1}
Print("Common element is", di)
```

Result: Program is executed and output is verified

Result

Enter the element with Spaces : 1 5 55 4 41

Common element is {'4', '1'}

Program 11 Date: 28-02-2021

Aim: Python Program to replace a Character in a String.

Program

```
String = input("enter the Character:")
re = input("enter the character to be replaced:")
new = input("enter new Character to be inserted:")
Print("new String is:", String.replace(re, new))
```

Result: Program is executed and output is verified

Result

Enter the Character: good morning

Enter the Character to be replaced: morning

Enter new Character to inserted: night

new string is : good night

Program 12

Date: 28-02-2021

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

Program:

```
String = input("enter the String:")
first = String[0]
last = String[-1]
ex = last + String[1:-1] + first
Print(ex)
```

Result: Program is executed and output is verified

Result

Enter the String : RAM

MAR

Program 13

Date: 28-01-2021

Aim: Python Program to merge two dictionaries.

Program:

```
dict1: {"name": "MAY"}

dict2: {"Date": 21}

dict1.update(dict2)

Print ("merged dictionarie is", dict1)
```

Result: Program is executed and output is verifyed

Result

merged dictionaries {'name': 'MAY', 'Date': 21}

Program 14 Date: 28-02-2021

Aim: Python Program to ascend and decent dictionary.

Program:

```
y={'carl':40,'alan':2,'bob':1,'danny':3}
l=list(y.items())
l.sort()
Print("Ascending Order is",l)
l=list(y.items())
l.sort(reverse=True)
Print("Descending Order is",l)
dict1=dict(l)
Print("Dictionary",dict1)
```

Result: Program is Executed and output is verifyed

Result

Asending order is:

[('alan', 2), ('bob', 1), ('carl', 40), ('danny', 3)]

Deseading order is:

[('danny', 3), ('carl', 40) ('bob', 1), ('alan', 2)]

Dictionary

{'danny': 3, 'carl': 40, 'bob': 1, 'alan': 2}

Program 15 Date: 28-02-2021

Aim: Python Program to remove even numbers from a list.

Program:

```
List1 = [1, 2, 3, 4, 5, 6, 7, 8]

out = []

for i in list1:

  if i%2 == 0:

    list1.remove(i)

    Out.append(i)

    Print("list of Odd numbers is", list1)
```

Result: Program is executed and output is verifyed

Result

list of odd number is [1, 3, 5, 7]

Program 16

Date: 01-03-2021

Aim: Python Program to find gcd of a number.

Program:

```
X = int(input("enter 1st number:"))
Y = int(input("enter 2nd number:"))
If X<Y:
   Sm = X
else:
   Sm = Y
for i in range(1, Sm+1):
   if X%.i ==0 and Y%.i ==0:
   gcd = i
Print("gcd of {0},{1} is:".format(X,Y),
                                  gcd)
```

Result: Program is executed and output is verifyed

Result

Enter 1st number : 54

Enter 2nd number = 24

The gcd of 54 and 24 is 6

Program 17

Date: 01-03-2021

Aim: Python Program to find factorial of a number.

Program:

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

fact = 1
for i in range(1, x+1):
    fact = fact * i
Print("factorial of {0} is".format(x),
                                fact)
```

Result: Program is verified and output is executed

Result

Enter a number : 5

factorial of 5 is 120

Program 18 Date: 01-03-2021

Aim: Python Program to find fibonacci Series

Program

```
terms = int(input("how many term:"))
n1, n2 = 0, 1
count = 0
if terms <= 0:
    Print("please enter a Positive integer")
elif terms == 1:
    Print("fibonacci Sequence up to",terms)
    Print(n1)
else:
    Print("fibonacci Sequence:")
    while count < terms:
        Print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1
```

Result

-> how many term: 0

Please enter a Positive integer

-> how many term: 1

fibonacci Sequence up to 1

0

-> how many term: 5

fibonacci Sequence:

0
1
1
2
3

Program 19 Date: 01-03-2021

Aim: Python Program to Perform String Functions

Program

```
String1 = " Apple"
Print (" initial String", String1)
Print(" First character of String:",
                String1[0])
Print(" last character of String:",
                String1[-1]).
```

Result: Program is executed and output is verified.

Result

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.

Program:

```
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)
```

Result: Program is executed and output is verifyed

Result

4624

6084

6400

8464

Program : 21

Display the given pyramid with step number accepted from user.

Program:

```
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()
```

Result: Program is executed and output is verified

OUTPUT

```
Enter the number of rows: 6
1
2 4
3 6 9
4 8 12 16
```

Program : 22

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

Program

```
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'))
```

Result: Program is executed and output is verified

Output

{'f': 1, 'a': 1, 'c': 2, 'e': 1, 'b': 1, 'o': 3, 'k': 1,

'!': 1, 'm': 1}

Program 23

Aim: Add 'ing' at the end of a given String

If it already end with 'ing' then add 'ly'.

Program

```
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('think'))
print(add_string('string'))
```

Result: Program is executed and output is verified.

Output

Cryingly

thinking

Stringly.

Program 24

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

Program

```
def longestLength(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 Seperated
    by Space")
a = a.split().
longestLength(a)
```

Result: Program is executed and Output is verified

Output

Enter a list of elements Seperated by Space India is my Country

The word with the longest length is 7

Program 25

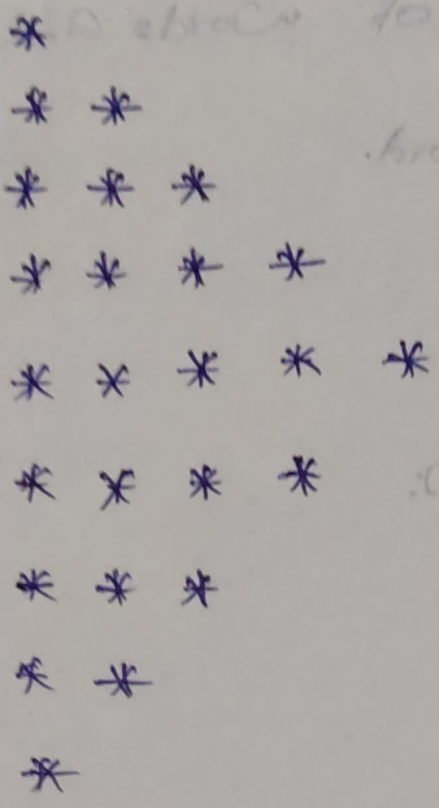
Aim: Construct following pattern using nested loop.

Program:

```
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(" ").
```

Result Program is executed and output is verified

Output



Program 26

Aim: Generate all factors of a number.

Program:

```
def Print_factors(x):
    Print("The factors of",x,"are"):
    for i in range(1,x+1):
        if x%i==0:
            Print(i)

num = 10
Print_factors(num)
```

Result: Program is executed and Output is verified

Output

The factors of 10 are

1

2

5

10.

Program : 27

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

Program:

```
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(10,20))
Print (" area of circle (15) is:", C_area (20))
Print (" area of triangle (12,20) is",
    t_area (22,20))
```

Result: Program executed and Output Verified

Output

Area of rectangle (30,20) is 200

Area of circle (15) is : 1256.6370614359173

Area of triangle (12,20) is 220.0

Program 28.

Aim: work with build-in Package

```
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
Print(x.Strftime("%x.%A"))
```

Result: The Program has been executed and the outcome has been verified.

Output

2021-03-20 ~~04:13:33.347291~~

04:13:33.347291

2021

Sunday.

Program : 29

Aim: Create a class rectangle with attributes length and breadth and methods to find area and Perimeter. Compare two rectangle Objects by their area.

Program:

```
class Rectangle:
    def __init__(self,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 1st rectangle:"))
b = int(input("enter breadth of 1st rectangle:"))
c = int(input("enter length of 2nd rectangle:"))
```

```
d = int (input (" enter breadth 2nd 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")
```

Result:

The Program was executed and the result was verified.

Output

Enter the length of I^st rectangle : 5

Enter breadth of I^st rectangle : 10

Enter length of 2^nd Rectangle : 7

Enter breadth of 2^nd rectangle : 10

Area of Ist rectangle : 50

Area of 2^nd rectangle : 70

Perimeter of 1st rectangle : 30

Perimeter of 2^nd rectangle : 34

not equal.

Program 30

Aim: Create a package graphics with module rectangle, Circle and Sub-package 3D graphics with modules cuboid and Sphere. Include method to find area and Perimeter of respective figure in each module. Write Program that find area and Perimeter of figure by different importing statements. (include Slective import of modules and import * Statements).

Program:

Circle.Py

```
From math import pi
 def (area (r):
   return (pi*r*r)
 def ( Perimeter (r):
   return (2*pi*r).
```

Retangle.Py

```
def rarea(l,b):
    return(l*b)
def rperimeter(l,b):
    return(2*(l+b))
```

Inside the folder of Graphics.Py create.
Another folder dgraphics.Py Sphere.py.

```
from math import pi
def Sarea(r):
return(4*pi*r*r)
def Sperimeter(r):
return(4/3)*pi*r*r*r)
```

Cuboid.Py

```
def Cuarea(l,w,h):
return(2*l*w)+(2*l*h)+(2*w*h))
def Cuperimeter(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", rarea(l,b))
Print("Perimeter of rect", rperimeter(l,b))
Print("Circle")
r = int(input("enter a radius"))
Print("Area of circle:", Carea(r))
Print("Peri of Sphere", Sperimeter(r))
Print("Sphere")
r= int(input("enter a radius"))
Print("area of Sphere:", Sarea(r))
Print("peri of Sphere:", Sperimeter(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", (Uarea (l, w, h))
Print ("Peri of cuboid", (Uperimeter (l, w, h))
```

Output

Rectangle:

Enter length: 5

Enter breadth: 10

Area of rectangle: 50

Perimeter of rectangle: 30

Circle

Enter radions of Circle: 4

Area of Circle: 50.265482 45

Perimeter of Circle: 25.132741778

Sphere

Enter radions of Sphere: 3

Area of Sphere: 113.097 33 55292 3255

Perimeter of sphere: 113.0973355 2923255

Cuboid

Enter length: 7

Enter width: 12

Enter height: 8

Area of Cuboid: 472

Perimeter of Cuboid: 104

Program 31:

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

Program

```
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 avilable balance=", self.balance)
    S = Bank_Account().
    S.deposit()
    S.withdraw()
    S.display()
```

Result

The Program was executed and the Output was verified.

Output

Enter Amount to be deposited : 900

Amount deposited : 900.0

Enter Amount to be withdrawn : 400

you withdraw : 400.0

Net Avilable 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.

Program:

```
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
rect1 = A(3,4)
```

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

Result

The Program was executed and the Output was verified.

Output

React 2 is greater than React 1.

Program : 33

Aim: Create a class time with private attributes hour, minute and second. Overload '+' Operator to find Sum of 2 time.

Program

```
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
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 format
        HH.mm.ss\n")
h1,m1,s1 = map(int, time1.split(':'))
```

t_1 = Time (h_1, m_1, s_1)

t_2 = Time (h_2, m_2, s_2)

$t_1 + t_2$.

Result

The Program was executed and the Output was verified.

Output

Enter first time in the format HH.mm.ss

2:2:3

Enter Second time 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.

Program:

```
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 books", "Training Python By
Programming", "Jeeva Jose", 200, 249)
S1.display()
```

Output

Title : Training 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

Program

```
Str1. "welcome to Python Programming"
      "\n" "Python" "\n"
fw = open ("A file.txt","w")
fw = write (str1)
fw = close ()
fr = open ("A file.txt","r")
Str2 = fr.readLines ()
for i in Str2:
    Print (i)
```

Result: The Program was execute and Output has been verifyed.

Output

welcom to Python Programing

Python

Program : 36

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

Program

```
F = open ("demo.txt", 'r')
Str1 = F.readLines()
f.close()
f = open ("file2.txt", 'w')
x = 0
for i in Str1
    x = x+1
If x%.2 != 0:
  f.write(i)
  fclose
f = open ("file2.txt", 'r')
Str2 = f.readLines()
Print (Str2)
```

Output: Program executed and output verified

Output

['how\n', 'are not beautiful\n', 'hehe']

Program 37

Aim: write a Python Program to read each row from a given csv file and Print a list of strings

Program

```
Import csv
with open ('email.csv', newline='') as csvfile
    data = csv.reader (csvfile, delimiter = ' ',
    quote char = '|')
    for row in data:
    Print (','.join (row))
```

Result: Program has executed and output is verified.

Output

Login email ; Identifier ; firstname ; Lastname

Raman@gmail.com ; 2070 ; Raman ; KA

Crai@gmail.com ; 4981 ; Crai ; Jhonson

Program 39

Aim: Write a Python Program to read specific Columns of a given csv file and Print the Content of the Columns.

Program:

```
Import, csv
with open("fruits.csv",'w',newLine='')as
                                    csvfile:

    write = csv.writer(File)

    write.writerow(["sino", "fruits","rate"])

    write.writerow(["1","apple","60"])

    write.writerow(["2","orange","55"]]

with open("fruits.csv",'r')as file:

    data = csv.reader(file)

    Print("Content in Colouns fruits:")

    for r in data:

        Print(r[1])
```

Result: Program executed and Output verified

Output

Contents in column 'fruits'

fruits

Apple

Orange

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 content

Program

```
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 = "temp.csv"
```

```
try:
with open(csv_file,'w') as csv file:
writer = csv.DictWriter(csv file, feild names=
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)
```

Result: Program is executed and output is Verified.

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'], 'column2': [33,25,56], 'column2':