

1960

to station for exhibition until next

1 FEB 1961

Python Lab Record.

3/20/61 Go exhibit with others

4 FEB 1961

All done. See you.

On my way back.

Arrived at

Brooklyn Center for the Performing Arts
5/14/61 28th St. Brooklyn, NY

Date

Event: Python exhibited and Outgoing
mailings

Program No:1

Date: 18-02-2021

1/1/2021

Aim: To find area of a circle

Program:-

```
import math
```

```
def Area_of_circle(r):
```

```
    a = r * r * 2 * math.pi
```

```
    return a
```

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

```
print("%0.2f" % Area_of_circle(r))
```

Output

Result: Program executed and Output

Verified

Result

Enter the radius of circle: 2

12.566371

Entered don't want it q

Enter the radius of circle:

28.274334.

Program No: 2

Date: 18-02-2021

Aim: To find largest among three numbers

26 ii c go snap2

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 verified

cos-10-81 contact

Result

enter the first number : 5

enter the Second number : 4

Enter the third number : 9

the largest is : 9

to evaluate and return "I am a program" and 9 as r

((+) shows -90 and X " 20, X") and 9

return

two two two last word report three

to print

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 903 is {13".format(a,Square)})
```

Result: Program has been Executed Output

Verified

Result of Q1

2007 import?

Enter the number: 5

Square of 5 is 25

((("return sum of two numbers") dsgn) sri = P

((("return twice of a number") dsgn) sri = d

((("return brint of a number") dsgn) sri = b

:((scr) bsr (scr) 2)

P = despal

:((scr) bsr (scr) 2) 110

d = despal

11010

b = despal

despal, "let despal sum") first

two no between import if less

key first

Program NO: 4

Date: 21-02-2021

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 verified

Result

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

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

import

("maximum value") diagonal = 10

$P \times P = \text{square}$

(max, 0) found. "10" is 10? To square?" invert

digital network and wait import lines

host

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 verified

Look Result: Final output

Enter the String : AmalJyothi

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

['I', 'H', 'E', 'S'] = set1]

: set1 ni or rot

[(n-a) dinirot

; output

['H', 'E', 'S', 'U'] = P

random & random] = contolVborwng?

[P ni random rot

(P ; contol Vborwng) print

(contolVborwng ; contol Vborwng) print

Output b/w between 2) of output : final

output

Date: 21-02-2021

Program : 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 verified

Result:

Enter the String : AmalJyothi College of Engineering

Kanjirappally.

String is : AmalJyothi College of Engineering

(Count of word is 5)

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 Verified

Result:

Enter the String: AmalJyothi

the String is : AmalJyothi

Count of a is : 1

String is AmalJyothi

(". point2 is null return") n/a = false;

(point2 == point2) n/a = true

(string2 - point2) n/a = error

(car, "a") break to two") n/a = true

length less than or equal to n/a = true

length of

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:: `list1`

length of list1 is 7

(`list1` contains 7 integers)

(": pointe est vrai") true = pointe

(pointe, "il pointe soit") soit

0 = vrai

: pointe est vrai

; $i = 1, 21$

(`for` (`i` `from` 1 `to` 20) `do`)

(`print` ("il est vrai")) vrai

two lines between `list1` and `list2`

list1 =

Program:9

Date: 27-02-2021

Aim: Python Program to check the sum of list.

[11, 12, 13] in forms of comma).

Program:

```
list1 = [1, 2, 34, 56, 78, 89, 76, 23]
```

```
total = sum(list1)
```

```
Print("Sum of list1 is:", total)
```

Result: Program is verified and output Executed

is verified

1006-2-10-210D

Result

Sum of list1 is : 359

list1 = [101, 102, 103, 104, 105]

$$[101, 102, 103, 104, 105] = 101 +$$

$$(101) \times 5 = 101 \times$$

$$(101, 102, 103, 104, 105) \rightarrow \text{map}(\lambda x)$$

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' }

[ES OF PR OF 22 HE 5 1] = 1421

(1421) mod 10101

(1421) mod 10101

right 6-10 leftmost 21 number 111101
between

leftmost 21

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

1506-10-FP str9

olmport

Result

Enter the Character: good morning

Enter the character to be replaced: morning

new st.

Enter new character to inserted: night

new string is : good night
("string")

Original str = v

if ((str[0] >= 'a' & str[0] <= 'z') || (str[0] >= 'A' & str[0] <= 'Z')) {

(str[0] = 'e' & str[0] == 'r' & str[0] == 'o' & str[0] == 'm' & str[0] == 'm' & str[0] == 'g')

digit has between olmport and begin of

begin of

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

II. output

Enter the String : RAM

MAR

prints out

Mar

Mar was printed to screen

(": character) and value") \rightarrow i = print

ad of mar was (": character) and value") \rightarrow i = 38

(": character

of char and value") \rightarrow i = 65

(": character ad of

((char, or) is also print, ("character and") print)

two bars between is output instead

before it

Program 13

Date: 28-02-2021

Aim: Python Program to merge two dictionaries.

`(c, fresh), (on, ice), (i, cool), (s, solo)`

Program:

`dict1 = {"name": "May", "city": "London", "age": 21}`

`dict2 = {"Date": 213}`

`dict1.update(dict2)`

`print("merged dictionary is", dict1)`

Result: Program is executed and output is verified

Bob Result

El map?

Merged dictionaries {
'name': 'MAY',
'age': 10, 'Date': '21'}

map?

(": print & write") + nqni = print

[0] print & write

[1] print & tool

string + [1:1] print & tool = x

(x) string

two bar between el map? length

length 27

Program 14

Date: 28-02-2021

Aim: Python Program to Ascend and descent 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 verified

Ques Result

Ans import

Ascending order is sorted using lambda

$\left[('alan', 2), ('bob', 1), ('carl', 40), ('danny', 3)\right]$

Descending Order is:

$\left[('danny', 3), ('carl', 40), ('bob', 1), ('alan', 2)\right]$

Dictionary

$\{('danny': 3, 'carl': 40, 'bob': 1, 'alan': 2)\}$

Two keys are inserted in import order
by first of

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 number(s)", list1)
```

Result: Program is executed and output is verified

Result

11 morphs

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

Breaking Order:

:junk', 'dog', 'c':nolo', on: 'moo' } = 4
(Dumb) (Dumb) (Dumb)
(Demand p) tail = 1

1 + 02 . 2

Dictionary

(l, "c":nolo probwesA") first

(Demand p) tail = 1

(CONP= server) first . 1

(l, "c":nolo probwesA") first

(Dumb) tail

(Dumb, 'pronounsA') first

two line between 21 morphs
last line 21

Program 16

Date: 01-03-2021

Aim: Python Program to find gcd of a number
number 3 to 1000000000

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 {} , {} is : ".format(X,Y),
```

```
gcd)
```

Result: Program is executed and output is verified

Result

21 morford

Enter 1st number: 54

Enter 2nd number = 24

The gcd of 54 and 24 is 6

[8, 6, 10, 11, 12, 13] = 12011

[] = 120

: 12011 at 1 no?

: 0 = 0 121 71

(i) answer. 1 121

(ii) answer. 120

(iii) (a) reduce 660 to 1011 " 121

lights has turned on in morford street

length of

Program 17

Date: 01-03-2021

Aim: Python Program to find Factorial

'of a number'. Output: 200

Program:

```
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)
```

Result: Program is verified and output
is executed

Result

Enter a number : 5

Factorial of 5 is 120

((("random" "ratns") (nq)) fnt) = X

((("random" "bsg" "ratns") (nq)) fnt) = V

: PDX 91

x = m?

: 2219

v = m?

((("m?" "1") (prn) v) ; rot

10 == 1Xv And 0 == 1.vX 91

i = b?P

((("b?" "1") (prn) v) ; rot

(b?P

is true or has between 11 numbers

Program 18

Date: 01-03-2021

Aim: Python Program to find Fibonacci

Series

Program

```
terms = int(input("how many terms:"))
```

```
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

Florinport

→ how many terms: 0? not float

please enter a Positive integer⁺

→ how many term: 1

Florinport

Fibonacci Sequence up to 1

0

1 = 1307

→ how many term: 5

(1307, 1) spawn of it not

Fibonacci Sequence:

1307 = 1307

(1307, 1) spawn of it not

1
1307
1
2
3

1307 has length of 1307 length

between 1

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

81 morport

initial string: Apple, 9 config: 01A

First character of string: A?

Rest character of String: e

config

((first part word) Engg) fair > emrpt

String Sequence 1, 0 = in, 1 =

0 = thru

: 0 = > emrpt 91

(rest of string word config "91") fair?

String Sequence 1, 0 = emrpt 91 5

(emrpt of rest string config "91") fair?

(in) fair?

0210

("engg is modif") fair?

: emrpt > thru 5 thru

(in) fair? 1

01 + 00 = 010

0010 = 001

1 + 0001

Program 20:

7UGND

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 verified

Result

PI monpoly

4624

6084

6400

8464

"S199A" = 1 point?

(1 point, 1 point testing) twist

1 point to rotation twist "I" twist

([0] 1 point)

1 point to rotation twist "C" twist

([0] 1 point)

1960 has between 21 monpoly mixed

aspirin d

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

right justified output is as follows

using cout < < endl; each row has 10 columns

3 6 9

4 8 12 16

: (1,00001,0001) space is not

: (1,001,158) space is (not)

: i+i = : 71

(i) right justified

= 2.36 (0 points) for loop 0 = -5.8 [0] points 12) for 91

(0) points for loop 0 = -5.8 [(c) points) for loop 0

0 = -5.8

(i) failing

right justified output is monospaced since

leftmost of

Program = 22

Topic

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

Program

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,
'v': 1, 'w': 1} most frequent words

76 - morph

((words + stem + affix) length) for each

(word, 1) spear will not

((1), 1) spear will not

(`t-h-s, t-x-i) first

77 - morph

if length - 2 is between 11 morph? if yes 9

begin

Program 23

depth

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

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

Program

def add_string(str):

length = len(str)

If length > 2:

If str[-3:] == 'ing':

str += 'ly'

else:

str += "ing"

return str

Print(add_string('crying'))

Print(add_string('think'))

Print(add_string('string'))

Result: Program is executed and output
is verified.

Output

Cryingly

thinking

Stringly

(Cry) for compensation

$$EB = 15.6$$

information

(Cry) for compensation

need of a 21

I = 4.7 bits

1525

I = 6.2 bits

bits control

((max. 1000 bits)) for compensation

length two between in import

beginning of

Program 24

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

Program

```
def longestLength(a):
```

```
    maxi = len(a[0])
```

```
    temp = a[0]
```

```
    for i in a:
```

```
        if (len(i) > maxi):
```

```
            maxi = len(i)
```

```
            temp = i
```

Print ("The word with the longest length is:

"temp," and length is", maxi)

```
a = input("Enter a list elements Separated  
by Space")
```

```
a = a.split()
```

```
longestLength(a)
```

Result: Program is Executed and Output is Verified

Output

Enter a list of elements Separated by
Space India is my country

The word with the longest length is

(cont) $= \text{India}$

: $\leq \text{India}$

: 'fun' = [: e - India . 21]

'fun' = + wh2

: $\leq \text{fun}$

'fun' = + wh2

+ wh2 contd?

(('verb') exist - bbd) + wh2

(('verb') print - bbd) + wh2

(('verb') print - bbd) + wh2

two lines between el morphs

: begin 21

Programs

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

* Diboco to tell address via

* *

* * *

* * * *

* * * * *

* * * * (0) Mysleep 20/

* * * (0) Aest = 1 XPM

* *

*

[0] P = 9m2f

in 0.1 sec

:(1XPM<(1)est) 91

(1)est = 1 XPM

2 sec

:> Mental leap of synthesis from salt " to salt "

(1XPM, "salt Mental leap", salt "

advise > change of tell a robot " (work) = 10

(" change of

(C) 11/19/20 - 10

Indirect

(0) Mental leap

Program & G

Aim: Generate all factors of a number.

Program:

```
def Print_Factors(x):
```

```
    print("The factors of {} are:".format(x))
```

```
    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

import

The factors of 10 are

1

2

5

10.

good basis

import

: (even, 0) spots at 1, 10

: (1+9, 0) spots at 1, 10

(1+1=2, "x") pair

(1+9) pair

: (1, 0, 1+even) spots at 10

: (1+1, 0) spots at 1, 10

(1+1=2, "x") pair

(0, 0) pair

27 figures have instances of import import

by hand

Program :- 7

Date _____

Aim: Write lambda functions to find area of
Square, rectangle and triangle for
15-05-2023

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(10,10))

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

38 marks?

Area of rectangle (30, 20) is 200

Area of Circle (15) is 1256.6370614859173

Area of triangle (12, 20) is 220.0

10. If "or" is used instead of "and" then

:(1+X, 1) appears in i not

$$O = 1 \times 91$$

(1, 3) using

O1 = min

(min) instead of -1.17

right has been added in mark since

before 21

Program 28.

Topic

Program: works 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("%Y.%A")).

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

Output

F 80 + report

2021 - 03 - 20

04:13:33, 347291

04:13:33.347291

2021

Sunday.

barometer & 19.11am : bar abdmal = zero - 1

stndrd: ind. abdmal = zero - 1

(top) zero - r'ci (os, os) aligned to zero") 10189

((os) zero - l'ci (os) aligned to zero") 10189

'ci (os, os) aligned to zero") 10189

((os, os) zero - t

light O has broken support . fine

light

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:"))
```

`a = int(input("enter length of rectangle:"))`

`breadth`

`obj = Rectangle(a,b)`

`obj1 = Rectangle(c,d)`

`print("Area of 1st rectangle:", obj1.area())`

`print("Area of 2nd rectangle:", obj1.area())`

`print("Perimeter of 1st rectangle is:",
 obj1.peri())`

`print("Perimeter of 2nd rectangle is:",
 obj1.peri())`

`If obj1.area() == obj1.area():`

`print("Equal")`

`Else:`

`print("not Equal")`

Result:

The program was executed and the result was verified.

Output

Enter the length of 1st rectangle: 5
(x) + 109

Enter breadth of 1st rectangle: 10
(x) + 109

Enter length of 2nd Rectangle: 7
(x) + 109

Enter breadth of 2nd rectangle: 10
(x) + 109

Area of 1st rectangle: 50
(x) + 109

Area of 2nd rectangle: 70
(x) + 109

Area
Perimeter of 1st rectangle: 30
(x) + 109

Perimeter of 2nd rectangle: 34
(x) + 109

not equal.
(x) + 109

(x) + 109

(x) + 109

(x) + 109

(("ax") + 109)

(x) + 109

(x) + 109

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 Selective 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 Spesimeter(r):
```

```
    return((4/3)*pi*r*r*r)
```

Cuboid . Py

```
def Carea(l,w,h):
```

```
    return(2*l*w)+(2*l*h)+(2*w*h))
```

```
def Cperimeter(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 Peri of Sphere!", sArea(r))
```

```
-Print("Peri of Sphere.", sPerimeter(r))
```

```
Print("Cuboid")
```

```
l = int(input("Enter a length"))
```

```
w = int(input("Enter a width"))
```

```
h = int(input("Enter a height"))
```

```
Print("Area of Cuboid", (Area(l,w,h)))
```

```
Print("Peri of Cuboid", (Perimeter(l,w,h)))
```

Output

Rectangle :

Enter length : 5

Enter breadth : 10

Area of rectangle : 50 and 0 unit²

Perimeter of rectangle : 30 Unit

Circle

Enter radius of circle : 4

Area of circle : 50.26548245

Perimeter of circle : 25.132741778

Sphere

Enter radius of sphere : 3

Area of sphere : 113.09733552923255

Perimeter of sphere : 113.09733552923255

all numbers are correct

. both are correct

Cuboid

Face diagonal opposite to short edge
Enter length: 37 mm; short side of abutment
width: 12 mm

Enter width: 12 mm

Enter height: 18 mm

Area of Cuboid: 472

Perimeter of Cuboid: 108

length of abutment to interior
of abutment by 12 mm
and abutment to frog of abutment
and abutment to frog of abutment

(translate & frog)

compart

pg. 312(1)

Calculate 19 frog & 4 mm mort

area of square: $(r \times r) \text{ mm}^2$

$(r \times r \times 19) \text{ mm}^3$

: $(r \times r \times 19) \text{ mm}^3$

$(r \times r + 2) \text{ mm}^3$

Program 31:

Aim: Create a Rectangle Bank account with members Account number, Name, type of Account and balance . with 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("Amount Deposited", amount)

def withdraw(self):

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

If self.balance > amount

self.balance = amount

Print("In you withdraw", amount)

else:

Print("In insufficient balance").

def display(self):

Print("In net available balance =", self.balance)

S = BankAccount()

S.deposit()

S.withdraw()

S.display()

Result

The Program was Executed and the Output was Verified.

Output ((first deposit) 100.0
Enter ((Amount to be) Deposited : 900
((Amount deposited : 900.0))

((Enter) Amount to be withdrawn: 400
((Amount withdrawn: 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.

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.
(React 2 contains more of the reactant)

Ans 2

(React 2 is different w/ ") Ans 2

(Ans 2) perhaps Ans 2

(React 2 is different from React 1 in w/) Ans 2

(1) React 1 is not equal to 2

(1) React 2

(2) React 1 is not equal to 2

(1) perhaps 2

Ans 2

both have increased over normal anti
radiation endogenous

radiation endogenous

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("."))

$t_1 = \text{Time}(h_1, m_1, s_1)$

length

$t_2 = \text{Time}(h_2, m_2, s_2)$

$t_1 + t_2$

Result

The Program was executes and the
Output was verified.

Output

Enter first time in the format HH.MM.SS
100<291

2:2:3

Enter Second time in the format HH.MM.SS
100<291

4:58:59

NEW TIME

7:11:2.

:00<m 21

:1+n=2

:00<m 21

:1+n=2

:00+n=21

printf("NEW TIME")

((e,m,n)format : 22:11:103)fscanf

Operator

twodigit sum four digit) twodigit = sum

("00.MM.HH"

sum of digits three digits) twodigit = sum

("00.MM.HH"

((.) three digit sum, three digit = 10, three digit)

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, 219)

S1.display()

Output

(Computer Lab Report)
Title : Training Python by Programming
Author : Jeena Jose

Price : 200

Lines of

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 verified.

Output

welcome to Python Programming

Python

(str1, str2, str3) = ("Hello", "World", "123")

(str1 + str2, str1 * 3) = ("HelloWorld", "HelloHelloHello")

((str1, str2), val) =

(("Hello", "World"), str1 + str2) = ("HelloWorld", "HelloWorld")

((len(str1) - len(str2), str1)) = (2, "Hello")

(len(str1), len(str2), str1[-3:-1]) = (5, 5, "ell")

str1[2] = str1[2] + str2[2]

str1[2:10:2] = str1[2:10:2] = "HelloWorl"

((str1, str2), val) = (str1, str2)

(str1, str2, str3) = ("Hello", "World", "123")

(str1 * 3, str2 * 3, str3 * 3) = ("HelloHelloHello", "WorldWorldWorld", "123123123")

(str1 * 3, str2 * 3, str3 * 3) = ("HelloHelloHello", "WorldWorldWorld", "123123123")

(str1 * 3, str2 * 3, str3 * 3) = ("HelloHelloHello", "WorldWorldWorld", "123123123")

print("Hello", "World", sep=" ") = Hello World

(10, 20, 30, 40, 50, "Programming") = 10 20 30 40 50 Programming

Program : 36

to gno

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

recent.txt (mailamp@ramb3)

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)
```

```
F.close
```

```
F = open ("file2.txt", 'r')
```

```
Str2 = F.readlines()
```

```
Print (Str2)
```

Output: Program Executed and Output Verified

Output

Return of mapobj, nothing to draw
 ["hasIn" are not beautiful in, hehe
 so that it will end until we will see]

mapobj

"unmapobj nothing to draw" - 128

"if" "nothing" "and"

("w", "ext.17 & ?) or 0 = w?

(128) draw + w?

(0 draw) + w?

("w", "ext.17 & ?) or 0 = w?

(0 draw) + w? = 0

: 0 is 0 is not

(i) draw

but should not mapobj off - 128

because now can not do

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('emails.csv', newline='') as csvfile  
    data = CSV.reader(csvfile, delimiter=',',  
                      quotechar='\'')  
    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

(rai@gmail.com; 4991; rai, Johnson
mpg09)

('w', 'ext. ext. 2117') mpg09 = ?

(2291929) 2117 = mpg

229012 = ?

('w', 'ext. ext. 2117') mpg09 = ?

o = X

128 = o! at i not

1AX = X

o = ! 2.107 11

(2291929) 2117 = ?

229012 = ?

('w', 'ext. ext. 2117') 229012 = ?

(2291929) 2117 = 229012

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
```

(CSV file:

```
write = csv.writer(file)
```

```
write.writerow(["s.no", "fruits", "rate"])
```

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

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

```
with open("fruits.csv", "w") as file:
```

```
data = csv.reader(file)
```

```
print("Content in Coloumn fruits:")
```

```
for r in data:
```

```
    print(r[2])
```

Result: Program Executed and Output Verified

Output

Contents in columns of Fruits

Fruits still very soft & most won't open until 2-3 days

Apple

Orange

Report

very fragrant

silvered (or polished, 'reflecting') very well

' = shiny/silky, still very soft when ripe & when

('Y' = round stones)

soft or very soft

((soft) now, 'Y') fruit

two to three weeks and report final
bottom of

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, 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)

Result: Program is executed And output is

Verified.

Output

{id: [1, 2, 3], 'columns': [33, 25, 56], 'column2':
 (1, 2, 3) want to make)

{id: [1, 2, 3], 'columns': [33, 25, 56], 'column2':
 (1, 2, 3) want to make)

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

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

{id: [1, 2, 3], 'columns': [33, 25, 56], 'column2':
 (1, 2, 3) want to make)

(1, 2, 3)

(1, 2, 3) want to make

(["star", "circle", "square"]) want to make

(["od", "bigg", "1"]) want to make

(["ee", "pmko", "o"]) want to make

: 3/17 20 (1, 2, 3) want to make

(1, 2, 3) want to make

(["circle", "square"] want to make)

: make it r o f

(E1 r) f i l l

keeping two line between morph field