

Date: 14/01/2021

Program No. 1

Aim: Python Program To find area

```
def area (r):
```

```
    pi = 3.14
```

```
    return pi*(r*r)
```

```
num = float (input ("Enter the value:"))
```

```
print ("Area is %.2f", area (num))
```

Result:

The program has been executed &
The output was verified.

Result:

The program has been executed &
The output was verified.

Output

Enter The Value: 3

Area is 28.2600

Program No 2

Aim: Python program to find largest among 3 numbers

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

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

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

If (number1>number2) and (number1>number3):

 largest=number1

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

 largest=number2

else :

 largest=number3

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

Result:

The program has been executed and the output was verified.

Output

Enter the first number: 2

Enter the second number: 4

Enter the third number: 5

The largest number is 5

Program No 3

Aim: Python program To find square of a number

```
digit = int(input("Enter an Integer numbers:"))
square = digit * digit
print ("Square of {digit} is {square}")
```

result:

The program has been executed and
The output was verified.

Output

Enter an integer. number 4
square of 4 is 16.

Date: 26/01/2021

Program No 4

Aim: Python program To find area of circle

From math import pi

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

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

Result:

The program has been executed and
the output was verified.

~~E off meadow~~

Output

Enter the radius of the circle : 4

The area of the circle with radius 4.0 is
50.2654

$$\text{Fifths} + \text{Sixths} = \text{Score}$$

{Score} = {Fifths} + {Sixths}

Score

breakdowns and meadow off

footpath cow fogbow off

Date: 26/01/2021

Program No 5

Aim: Python program to find square of n

list1 = [14, 20, 13, 8, 6, 2]

for n in list1:

 square = n * n

 print(n, "squared is", square)

Result:

The program has been executed and
The output was verified

Report

14 squared is 196

20 squared is 400

13 squared is 169

8 squared is 64

6 squared is 36

2 squared is 4

Date: 26/01/2021

Program No: 6

Aim: Python program To find vowels in string

String1 = "Hello... how are you"

Point (Given string : In", string1)

vowels = "AaEeIiOoUu"

res = set([each for each in string1 if each in vowels])

point ("The vowels present in the string : In", res)

Result:

The program has been executed and
the output was verified.

Diphthong

Goren Stomg :

"Hello... How are you?" [ə, ʌ, ɒ, ʊ]

The vowels present in the stomg:

{'ʊ', 'ʌ', 'e', 'ɒ'}

(coupe, "coupe", m) Diphthong

grouped together

Date: 26/01/2021

Program No: 7

Aim: Python program to count words in a sentence

```
def word_count(s):  
    count = dict()
```

```
    words = s.split()
```

for word in words:

```
    if word in count:
```

```
        count[word] += 1
```

```
    else
```

```
        count[word] = 1
```

```
return count
```

```
print(word_count("When you change the  
quality of your thinking, you change the  
quality of your life sometimes instantly"))
```

Result:

The program has been executed and
the output was verified.

Output

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

Date: 26/01/2021

Program No: 8

Aim: Python program to count a mark?

a = ['anto', 'seban', 'roshan', 'joseph']

str1 = (''.join(a))

count = 0

for i in str1:

If i == 'a':

count = count + 1

print ("Count of 'a' in the list is :" + count)

Result:

The program has been executed and
the output was verified.

Output

cost of 'a' make bus is: "3000/-"

With respect

(a) Price of bus per km

(b) Fuel expenses

(c) Salaries etc. & Bus

expenses on drivers

Prices of buses: ₹

1 = [bus] fares

1 = [bus] fares per km

1 = fares per km

1 = inputs per km = (Fares + Bus) / km

1 = inputs per km = fare per km + Bus

1 = fare per km + Bus

Date: 26/01/2021

Program No 9

Aim: Python program to check the length of list

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

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

len1 = len(list1)

len2 = len(list2)

If len1 == len2:

print('both list have equal length')

else:

print('both list doesn't have equal length')

Result -

The program has been executed and
the output was verified.

Output

both list have equal length.

[class] [code] [code] [class]

((class)) is not

object

not an object

: class

it has = class

(class + "a" + class + "b" + "c" + "d")

but both are not objects

Date: 26/10

Program No:10

Aim: python program to check the sum of lists.

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

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

total1 = sum(list1)

total2 = sum(list2)

If total1 == total2:

print('both list have equal sum')

else:

print('both list have different sum')

Result:

The program has been executed and the output was verified.

class robal

Output

both list have equal sum

[51, 51, 51, 51, 51, 51, 51, 51, 51]

[51, 51, 51, 51, 51, 51, 51, 51, 51]

(150) not -

(600) not - c

: S not = 1 not

("digit loops and tail died") bar

("digit loops and hazards tail died") front

One has been used and mapped
baseline now right

Date: 27/01/2021

Program No: 11

Program

Ques: Python program to check the common element in the lists.

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

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

for value in list1:

 if value in list2:

 common = 1

 if common == 1

 print("there are common elements")

else

 print("no common elements")

Result:

The program has been executed and the output was verified.

Output

These are common elements

$$[E1, D1, Z1, A1, M1, E1, H1, C1, D1, O1] \rightarrow 1B_{all}$$

$$[C1, O1, S1, H1, O1, A1, Z1, H1, E1, C1, D1] = 2B_{all}$$

$$(1B_{all})_{moe} = 1b_{all}$$

$$(2B_{all})_{moe} = 2b_{all}$$

$$1b_{all} = 1B_{all}$$

more jumps and less diff. \rightarrow longer

Common smoothness and less diff. \rightarrow longer

Date: 27/01/2021

Program No: 12

Aim: Python program to replace a character.

```
def change - char(str):
    char = str[0]
    str1 = str.replace(char, '$')
    str1 = char + str[1:]
print(change - char(refresh))
```

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

order

deflesh

Non ~~meat~~

meatless really isn't
that it's not

[C1, A1, R1, D1, N1, E1, S1, G1, H1, O1, P1] = 161

[C1, P1, R1, D1, N1, H1, S1, G1, I1, O1, C1] = 161

: 161 or 160

: 6 fold at center H

H = common

H = common

(Chamela communis see Scott) found

(Chamela communis see Scott) found

floral

long tubular red and orange all
yellow now high up

Program No: 13

Program

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

```
def change_string(str1):
    return str1[-1:] + str1[1:-1] + str1[:1]
```

```
print(change_string('pineapple'))
```

Result:

The program has been executed
and the output was verified.

Output

eineapplp

: (Note) note - sprach

[Note] = note

((E>Note) note . note = 1 note

E: E>Note + note = 1 note

((deadbeats) note sprach) freq

leben und mitmachen ab fliegen
leben zwei Hesse 916 kris

Date: 27/01/2021

Program No: 14

Aim: Python program to merge 2 dictionaries

def Merge(dict1, dict2):

return dict2.update(dict1))

dict1 = {'a': 10, 'b': 8}

dict2 = {'d': 5, 'c': 2}

print(Merge(dict1, dict2))

print(dict2)

Result:

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

Odepu

{ d:5, c:2, a:10, b:8 }

E1 - 01 mmopco

prate in m' hotel Cost bus Col

: (note) prate - sprat

[1] note + [1-1] note + [1-1] note mope

((sluggish)) (prate - sprat) hump

: float

prate need eat mope sat

version now Rughu self - pane

Program No:15

Aim: Python program To ascend and descent dictionary

Import operator

d = {1:2, 3:4, 4:3, 2:1, 0:0}

print("Original dictionary : ", d)

sorted-d = sorted(d.items(), key=operator.itemgetter(1))

print("Ascending : ", sorted-d)

sorted-d = sorted(d.items(), key=operator.itemgetter(1), reverse=True)

print("Descending : ", sorted-d)

Result:

The program has been executed and
The output was verified.

Output

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

Dictionary in Ascending

[0: 0), 1: 2), 2: 3), 3: 4)

Dictionary in Descending

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

: Ready

Date: 27/11/2021

Program No: 16

Aim: Python program to remove even number from the list

list = [11, 22, 33, 44, 55, 66, 77, 88, 99]

```
print(list)
```

```
for i in list:
```

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

```
        list.remove(i)
```

```
print("list after removing:", list)
```

Result:

The program has been executed and the output was verified.

odd

[11, 22, 33, 44, 55, 66, 77, 88, 99]

16) after removing : [11, 33, 55, 77, 99]

remove 11

{0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

(11, (remove 11)) func

(remove 11, 11) func = 10 - 11 = 9

(11, 11, (remove 11)) func

(remove 11, 11, 11) func = 10 - 11 - 11 = 9 - 11 = -2

(11, 11, 11, (remove 11)) func

remove 11, 11, 11, 11 func
remove 11, 11, 11, 11, 11 func

Date: 27/01/2021

Program No: 14

Aim: Python program to find gcd of number

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

a = 45

b = 65

if (gcd(a,b)):

print('GCD of', a, 'and', b, 'is', gcd(a,b))

else:
 print('not found')

Result:

The program has been executed and
The output was verified

QDP?

600 of 45 865 is 5

[PP, 22, FF, 22, RR, NN, EE, LL, "]. Q

[Gen] fm

"Bell at i

(C = 2G + i) f

(i) Dromos, Bell

(Bell": Enron to Bell") f

Date: 03/02/2021

Program No.: 18

Aim: Python program to find factorial of a number

num = int(input("Enter a number"))

factorial = 1

If num < 0:

print("Sorry, factorial does not exist for the negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

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

factorial = factorial * i

print("The factorial of", num, "is", factorial)

Result:

The program has been executed and the output was verified.

Output

Enter a number 5

The factorial of 5 is 120

• (d, o) loop - for

• (d = > o) if

• while

(d >= o) loop numbers

end = 0

end = 0

• ((d, o) loop) li

loop, let d, & b, if and C for eq

(loop for) loop

to make a new end message it
will have one call of

Date: 03/02/2021

Program No 19

Aim: Python program to find fibonacci sequence

```
def recur-fibo(n):
```

```
    if n==1:
```

```
        return n
```

```
    else:
```

```
        return (recur-fibo(n-1)+recur-fibo(n-2))
```

```
nterms = int(input("How many terms?"))
```

```
If nterms <= 0:
```

```
    print("please enter a positive number")
```

```
for i in range(nterms):
```

```
    print(recur-fibo(i))
```

Result:-

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

Ques?

How many teams? 4

Fibonacci Sequence:

0
1
1
0

Program No:20

Aim : python program to perform string function.

def add_string(str1):

length = len(str1)

If length > 1:

If str1[-3:] == "ing":

str1 += 'ly'

else:

str1 += "ng"

return str1

print(add_string('do'))

print(add_string("according"))

Result :

The program has been executed and
the output was verified

out put

clown

accordingly

Date: 03/08/2021

Program 21

Aim: Python program to perform the sum of given items.

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

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

```
print(sum)
```

Result: This program has been executed and was verified

21

Output

Output m

22

reaction of monomer with

(rate) protobioles
(rate) rate of depol

: 1 < depol - 21

"Bal" = [E-] , other bi

V = + 1000

E_m = + 5000

rate number

Program No: 22

Aim: python program to find perfect even square number in a range

```
num1 = int(input("Enter a number: "))
num2 = int(input("Enter a number: "))
```

```
for i in range(32, 100+1):
    for j in range(num1, num2+1):
        if i == j*j:
            string = str(i)
            if int(string[0])%2 == 0 and
               int(string[1])%2 == 0 and
               int(string[2])%2 == 0 and
               int(string[3])%2 == 0:
                    print(i)
```

Result:

The program has been executed and the output was received.

Output

Enter a number : 4444

Enter a number : 9999

4624

[1, 6, 2, 2, 18, 6, 17]

6084

6300

(maximum) mod

8464

(min)

Date: 03/02/2021

Program No: Q3

Aim: Python program to display the given pyramid with step numbers accepted from user

```
lines = int(input("Enter a number:"))
```

i=1

j=1

```
while i <= lines:
```

d = 1

```
while j <= i:
```

temp = i * j

```
print(temp, end=' ', flush=True)
```

```
print(" ", end=' ', flush=True)
```

j = j + 1

```
print(),
```

j = j + 1

Result:

The program has been executed and the output was verified.

23

Output

1

2

3

4

5

6 7 8 9 10 11 12 13 14 15 16

: (1001, 68) Output on 1
(68 min 1 min) Output on 6 7 8 9
6 * 6 = 36 61

(i) α to β parrot

1000 0 = 0 1 (07 parrot) 6st 11

1000 0 = 0 1 (07 parrot) 6st

1000 0 = 0 1 (07 parrot) 6st

1000 0 = 0 1 (07 parrot) 6st

(i) Parrot

Program No: Q4

Aim: python program to count the number of characters in a string

def char_frequency(str1):

dict = {}

for n in str1:

key = dict.keys()

If n in key:

dict[n] += 1

else:

dict[n] = 1

return dict

print(char_frequency("helloworld you"))

Result:

The program has been executed and the output was verified

EE: 0.11 m approx

Oviparid

$\Sigma h: 2, e: 2, l: 2, \theta: 3, w: 1, a: 1, r: 1,$
 $y: 1, u: 13$

((maximum width) \times length) \times 1 = 29 ml

$t = i$

$t = j$

normal $\Rightarrow i$ slides

$t = b$

$i \Rightarrow j$ slides

$i \times b = gmat$

(normal \Rightarrow desk, " \Rightarrow bus, gmat) firing

(normal \Rightarrow desk, " \Rightarrow bus, " ") firing

$t \times b = b$

" (bus) firing

$t \times i = i$

blown

Program No: Q5

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

def find (word):

w1 = []

for n in word:

w1.append (len(n), n)

w1. sort ()

result = w1[-1][0], w1[-1][1]

print ("longest word:", result[1])

print ("length of the longest word:", result[0])

find ["hello", "morning", "hi"]

Result:

The program has been executed and output was verified

Object

longest word: moon my

length of the longest word, 7

(size) ~~number of words - 200~~ 76

E3 - Posts

: wrote in a book
(size) ~~length - 600~~ 6

: post in a bl

E = [n] Posts

• sets

1 = [n] Posts

Posts numbers

(long horizontal) ^{*}) (horizontal) lines

Program No: Q6

Aim: python program to construct pattern using nested loop

ded star()

n = 5

for i in range(5):

 for j in range(i):

 print('*', end="")

 print("")

for i in range(n, 0, -1):

 for j in range(i):

 print('*', end="")

 print("")

star()

Result:

The program has been executed and the output was verified.

Output

*

X X

* * *

* * * *

* * * * *

* * * *

* * know me as rob

* * (a, (a) not (b) w = w

*

L7 = IVS

CD Room 2 top

L7G-31W L7D-31W = Class

(L7H-31W "know angel") Class

(L7H-31W Angel of the angel") Class

(L7H-31W "Oscarom" off at 1 hour)

Date : 03/02/2021

Program No : 24

Aim : Python program to print factors of a number

def print_factors(x):

print ("The factors of", x, "are :")
for i in range(1, x+1):

If x % i == 0:

print (i)

print-factors(232)

Result :

The program has been executed and the output was verified.

27

Output

The factors of 232 are:

1

2

4

8

29

58

116

232

28 : all multiples

multiple factors of multiplying nothing with
232 are: 1, 2, 4, 8, 29, 58, 116, 232

∴ note below

$$c = a$$

(a) factors of 232

(b) factors of 232

("," *) factors

("," *) factors

("," *) factors of 232

("," *) factors of 232

("," *) factors

("," *) factors

Program No: Q8

Aim: Python program to write lambda function
to find area of square, octangle & triangle.

```

point ("Enter the length of a side of square: ")
s = int (point ("Enter your value : "))

point ("Enter the length of rectangle & breadth ")
l = int (input ("Enter length: "))
b = int (input ("Enter breadth"))

point ("Enter base and height of triangle")
ba = int (input ("Enter base value : "))
h = int (input ("Enter height value!"))

x = lambda s: s*s
y = lambda l,b: l*b
z = lambda ba,h: ba*h*0.5

point ("Area of square is : ", x(s))
point ("Area of octangle is : ", y(l, b))
point ("Area of triangle is : ", z(ba, h))

```

Result:

The program has been executed and the output was recorded.

Output

Enter the length of aside of square

Enter your value : 2

Enter the length & breadth of rectangle

Enter your value : 4

Enter your value : 2

Enter the base and height of triangle

Enter your value : 3

Enter your value : 2

Area of square : 4

Area of rectangle : 8

Area of triangle : 3.0

Date : 17/02/2021

Program No : Q9

Aim: Python program to display future leap year from current year to a final year entered by user.

Import datetime.

a = datetime.datetime.now()

c = int(a.year)

b = int(input("Enter final year :"))

print("In leap years :")

for i in range(c, b+1):

if (i % 4 == 0):

print(i)

Result :

The program has been executed and the output was recorded.

29

Output

Enter final year: 2040

Leap year

2024

2028

2032

2036

2040

Exercise abdomen = 5
d x) d J abdominal = 5

Cardio Pulm abdominal = 5

(easy)
(hard)

Date: 17/02/2021

Program No: 30

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

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

pos = list1

for i in list1:

if i > 0:

pos.append(i)

print("Original list:", list1)

print("Positive Integer list", pos)

Result:

The program has been executed and the output has been verified

Output

PB - OK message

Original list: $[1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -67]$

Positive list: $[1, 2, 9, 87, 24]$

• remove symbols from list

• remove symbols from list = 10

(loop-i) for = 0

(i = 0 to 9) if (list[i] < 0)

(": append list[i] to list")

(i = 0 to 9) for (i = 0 to 9)

(i = 0 to 9) if (list[i] < 0)

(i = 0 to 9) if (list[i] >= 0)

(i = 0 to 9) if (list[i] >= 0)

Date : 17/02/2021

Program No : 31

Aim : python program to find biggest of 3 numbers entered.

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

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

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

If $a > b$ and $a > c$:

print(a, "is the biggest number")

elif $b > c$:

print(b, "is the biggest number")

else:

print(c, "is the biggest number")

Result :

The program has been executed and the output was verified.

Output

Output message

Enter 1st number : 5

Enter 2nd number : 6

Enter 3rd number : 8

8 is the biggest number

Read no. is 8

0 < i < 3

(i) largest. even

(Giant "Pai Lang Foo") Even

(Giant "Pai Lang Foo") Even

After the process, read and message all numbers greatest first

Date : 17/02/2021

Program No: 32

Aim : Python program to create a list of colors from comma-separated color names entered by user. Display first & last color

colors = input ("Enter colors separated by commas: ")).split (",")

print ("first color:", colors[0])

print ("last color:", colors[len(colors)-1])

Result:

The program has been executed & the output was verified.

Output

Enter colors separated by commas; red,
black, yellow.

Frost color : red

base color : yellow

Program No: 33

Aim: Python program to print out all colors from list not contained in color-list2

```
colors1 = set((input("Enter colors separated by commas : ")).split(","))
```

```
colors2 = set((input("Enter colors separated by commas : ")).split(","))
```

```
print("Colors in color-list1 not contained in color-list2 are : ", list(colors1.difference(colors2)))
```

Result :

The program has been executed and the output was verified

E : algorithm able to repeat entries
E : algorithm able to ignore entries

Output

Enter colors separated by commas: red,yellow,
brown

Enter colors separated by commas: black,white,

Colors in color-list1 not contained in color-list2

are: ['brown', 'red', 'yellow'] - colors

(",") filas. ((":) : commas

(["colors","":colors first]) frog

([i-(colors)not]colors,"":colors last") frog

sthaloRwes mod set moopong uti
Not last row bugRe

Date : 17/02/2021

Program No : 34

Aim : Python program to create a package graphics with modules rectangles, circle & sub packages 3D graphics with modules cuboid and sphere. Include methods to find area & perimeter of respective figures in each module. write programs that find area and perimeters of figures by different importing statements.

Circle.py

def area(r):

print('Area of circle with radius', r, 'is',
'1.2f' +. (3.14 * r * r), 'sq. units')

def perimeter(r):

print('Perimeter of circle with radius',
r, 'is', '1.2f' +. (3.14 * 2 * r), 'units')

Rectangle.py

def area(a, b):

point ('Area of parallelogram with sides', a, 'and', b, 'is:', '1.2f' + (a * b), 'sq.units')

def perimeter(a, b):

point ('Perimeter of parallelogram with side', a, 'and', b, 'is', '1.2f' + (2 * (a + b)), 'units')

sphere.py

def area(r):

point ('Area of sphere with radius', r, 'is', '1.2f' + (4 * (3.14 * r * r)), 'sq.units')

def perimeter(r):

point ('Perimeter of (great circle of) sphere with radius', r, 'is', '1.2f' + (2 * 3.14 * r), 'units')

cuboid.py

def area(l, b, h):

point ('Total surface area of cuboid with dimension:', l, b, h, 'is', '1.2f' + (2 * ((l * b) + (b * h) + (l * h))), 'sq.units')

def perimeter(l, b, h):

 print('perimeter of cuboid with dimensions',
 l, b, h, 'is :', 4 * af -> (h * (l + b + h)), 'units')

Find perimeter.py

import circle

from rectangle import *

from graphics import graphics
 from 3D-graphics import cuboid,
 sphere

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

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

rectangle(a, b)

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

circle(r)

f = float(input('Enter length of the cuboid: '))

b = float(input('Enter width of the cuboid: '))

cuboid(f, b)

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

sphere(r)

Find Area.py

```
import circle  
from rectangle import *  
from graphics3D.graphics import cuboid, sphere
```

```
a = float(input("Enter the length of the rectangle"))  
b = float(input("Enter the breadth of the rectangle"))
```

```
area(a,b)
```

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

```
circle.area(r)
```

```
l = float(input("Enter the length of the cuboid :"))
```

```
b = float(input("Enter the breadth of cuboid :"))
```

```
h = float(input("Enter the height of cuboid :"))
```

```
cuboid.area(l,b,h)
```

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

```
sphere.area(r)
```

Result :

The program has been executed and the output was verified.

Output

Enter length of octangle : 4

Enter breadth of the rectangle : 3

Perimeter of octangle with sides 4.0 & 3.0
is : 14.00 units

Enter the radius of the circle : 2

Circumference of circle with radius 2 is 12.56

Enter length of the cuboid : 5

Enter breadth of the cuboid : 4

Enter height of the cuboid : 3

Perimeter of cuboid with dimensions 5.0, 4.0,
3.0 is 48.00 units

Enter the radius of the sphere : 2

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

Enter length of the octangle : 2

Enter length of the octangle : 3

Area of octangle with sides 2.0 and 3.0^{15}
 6.00 sq. units

SL.2

Enter the radius of the circle: 4

Area of circle with radius 4.0 is 50.24 sq.units

Enter length of the cuboid: 4

Enter breadth of the cuboid: 2

Enter height of the cuboid: 2

Total surface area of cuboid with dimensions 4.0, 2.0, 2.0 is 100. sq.units

Enter the radius of the sphere: 1

Area of sphere with radius 1.0 is 12.56 sq.units

pg. 10

(a) πr^2 (b)

(c) $4\pi r^2$ (d) $2\pi r^2$

(e) $2\pi r^2$ (f) $4\pi r^2$

(g) πr^2 (h)

(i) $2\pi r^2$ (j) $4\pi r^2$

(k) $2\pi r^2$ (l) $4\pi r^2$

pg. 11

Program No: 35

Aim: Python program To create a rectangle class with attributes length and breadth & methods to find area and perimeter.
Compare Two rectangle object by their area

class Rectangle

def __init__(self, l, b):

self.length = l

self.breadth = b

def area(self):

return self.length * self.breadth

def perimeter(self):

return (self.length + self.breadth)

def cmp(self, obj):

If self.area() > obj.area():

print('Rectangle with length = ', self.length, 'and',
'breadth = ', self.breadth, 'has the greater area')

elif self.area() < obj.area():

point ('Rectangle with length = ' obj.length, ' and
breadth = ', obj.breadth, ' has the greater
area')

else

point ('Both have equal area')

21. rectangle (9, 3)

22. rectangle (3, 2)

23. com (22)

Result:

The program has been executed and the
output was verified.

Ques

(PQ. No. 1001)

Rectangle with length 9 and breadth 3 has the greater area.

Reasons: (a) $9 \times 3 = 27$ and $3 \times 9 = 27$ both areas are equal.

(b) $9 > 3$

Reasons: (c) $9 > 3$

(d) $9 > 3$

Reasons: (e) $9 > 3$

Reasons: (f) $9 > 3$

Reasons: (g) $9 > 3$

(h) $9 > 3$

Reasons: (i) $9 > 3$

(j) $9 > 3$

Reasons: (k) $9 > 3$

Reasons: (l) $9 > 3$

Program No: 36

Aim: Python program to create a Bank account with member account number, name, type of account & balance. write constructor and methods to deposit at the bank and withdraw an account from the bank

Class Bank Account:

```
def __init__(self, a, n, t, b):
```

```
    self.acno = a
```

```
    self.name = n
```

```
    self.type = t
```

```
    self.bala = b
```

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

```
    self.bal + = a
```

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

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

```
If self.bal >= a:
```

```
    self.bal - = a
```

point ('Rs', a, 'withdrawn: current balance
Rs.', self.bal)

else :

point ("Insufficient balance to make this
transaction")

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

b = input ('Enter name of the account holder:')

t = input ('Enter account type:')

h = float(input ('Enter your balance:'))

aci = Bank account (a, b, t, h)

aci.deposit(float(input ('Enter amount to
deposit:'))))

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

Result:

The program has been executed and the
output verified.

Output

Enter account number: 0090090909090909

Enter name of the account holder: John

Enter account type: savings,

Enter your balance: 100000

Enter amount to deposit: 300000

Rs.300000.0 deposited! current balance

Rs. 400000.0

Enter amount to withdraw: 5000

Rs. 5000.0 withdrawn! current balance
is Rs. 395000.0

Date: 17/02/2021

Program No: 37

Aim: python program to create rectangle class with attributes length and breadth and methods to find area and perimeter, compare 2 rectangle object by their area.

Class Rectangle:

def __init__(self, l, b):

self.length = l

self.breadth = b

def area(self):

return self.length * self.breadth

def perimeter(self):

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

def comp(self, obj):

If self.area() > obj.area():

print('Rectangle with length = ',

self.length, 'and', 'breadth = ', self.

breadth, 'has the greater area')

if self.area() < obj.area():
 print('Rectangle with length = ', obj.length,
 'and breadth = ', obj.breadth, 'has the
 greater area!')

else:

print('They have equal area!')

r1 = Rectangle(2, 3)

r2 = Rectangle(3, 4)

r1 = comp(r2)

Result:

The program has been executed and
 the output was verified.

Program No. 38

Aim : python program to create a class rectangle with private attribute length & width. overload '' operator to compare the area of 2 rectangles.

Class Rectangle :

def __init__(self, l, w):

 self.length = l

 self.width = w

 self.area = self.width * self.length

def __lt__(self, other):

 if self.area < other.area:

 print('rectangle with length = ',

 self.length, 'and width = ', self.width, '

 has the lesser area !')

 elif other.area < self.area :

 print('Rectangle with length = ', other.length, 'and width = ', other.width, 'has the
 lesser area !')

else:

print('They have equal area!')

$l = \text{float}(\text{input}(\text{'Enter length of 1st rectangle: '}))$

$w = \text{float}(\text{input}(\text{'Enter width of 1st rectangle: '}))$

$R_1 = \text{Rectangle}(l, w)$

$l = \text{float}(\text{input}(\text{'Enter length of 2nd rectangle: '}))$

$w = \text{float}(\text{input}(\text{'Enter width of 2nd rectangle: '}))$

$R_2 = \text{Rectangle}(l, w)$

$R_1 < R_2$

Result:

The program has been executed and
the output was verified

Output

Enter length of 1st rectangle: 7

Enter length of 2nd rectangle: 8

Enter length of 1st rectangle: 9

Enter width of 2nd rectangle: 7

They have equal area!

Date : 17/02/2021

Program No. 39

Aim: Python program to create a class publisher(name). Desire class Book from publisher with attributes title and author.

Desire class Python from Book with attributes price & no-of-pages. write a program that display information about a python book.
use base class constructor invocation & method overriding

Class Publisher:

```
def __init__(self, name):  
    self.name = name  
  
def show(self):  
    pass
```

class Book(Publisher):

```
def __init__(self, title, author, name):  
    self.title = title  
    self.author = author  
  
    publisher.__init__(self, name)  
  
def show(self):  
    pass
```

class python (Book):

def __init__(self, p, no, title, author, name):

self.price = p

self.no-of-pages = no

Book.__init__(self, title, author, name)

def show(self):

print('Book title:', self.title)

print('Author:', self.author)

print('Publisher:', self.name)

print('price:', self.price)

print('No of pages:', self.no-of-pages)

P1 = python(565.90, 259, 'programming with
python', 'Gv Rossum', 'ABC Books')

P1.show()

Result

The program has been executed
The output was recorded

Output

Book Title : Programming with python

Author : GV Rossum

Publisher : ABC Books

Price : 565.9

No of pages : 250

39 > 3

Know how to read and manipulate data
variable can take any type of

Date: 21/02/2021

Program No: 40

Aim: python program to read a file line by line & store it into a list.

def file_read(fname):

with open(fname) as f:

c = f.readlines()

print(c)

file_read('demo.txt')

Result :

The program has been executed and the output was verified

Output

(steel) rolling stocks

: Common railings, rolling stock, H2O - Pneumatic

[A trailer is a vehicle designed for carrying bulk material, often on building sites. In, they are distinguished from dump trucks by configuration, a clamped]

(steel) water tank

(steel-H2O, steel steel) pump

(steel-pump, : steel) pump

(steel-H2O, 'building') pump

(steel-H2O, 'pump') pump

(steel-on-H2O, 'copper pipe') pump

(steel (mining) H2O, O.P. 200) rolling - 19
mining, mining vehicles, rolling

(water - 19)

Plastics

plastics and plastic materials etc

plastics and plastic materials etc

Date: 21/02/2001

Program No: 41

Aim: Python program To copy odd lines of one file To other

```
a = open('demo.txt', 'r')
```

```
b = open('t.txt', 'w')
```

```
c = a.readlines()
```

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

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

```
        b.write(c[i])
```

```
else:
```

```
    pass
```

```
b.close()
```

```
b = open('t.txt', 'r')
```

```
d = b.read()
```

```
print(d)
```

```
a.close()
```

```
b.close()
```

Result

The program has been executed and the output was verified

Dumper

They are distinguished from dump truck by configuration: a dumper is usually an open 4-wheeled vehicle with the load skip in front of the driver.

(a) Front

("front-end") dump-truck

Date: 21/02/2021

Program No = 42

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

Import csv

```
with open ('temp.csv', newline= '') as csvfile:  
    d = csv.reader(csvfile, delimiter = ',')  
    qvotedar = ('')  
  
    for s in d:  
        print (',', join(s))
```

Result

The program has been executed &
The output was recorded.

Digitized

$[1, 5, 3]$, $[33, 25, 5]$, $[35, 30, 30]$,
 $[21, 40, 55]$, $[71, 95, 55]$, $[10, 10, 40]$,
 $[1, 2, 3]$, $[33, 25, 5]$, $[35, 30, 30]$,
 $[21, 40, 55]$, $[71, 95, 55]$, $[10, 10, 40]$

(ii) $\text{S} = \{1, 2, 3\}$

$(1, 2)$ follows d

$1, 2, 3$
2409

$(1, 2)$ follows d
 $(1, 3)$ follows d

$(2, 3)$ follows d

$(1, 2, 3)$

Date: 21/02/2021

Program No: 43

Aim: Python program to read specific columns of a given csv file and print the content of the columns.

Import csv

```
with open('temp.csv', newline='') as csvfile:  
    data = csv.DictReader(csvfile)  
    print('ID NAME')  
    for row in data:  
        print(row['id'], row['column1'])
```

Result:

The program has been executed and the output was recorded.

Digit

10 Name

[1, 2, 3] [33, 25, 26]

[1, 2, 3] [33, 25, 56]

[1, 2, 3] [33, 25, 56]

[1, 2, 3] [33, 25, 56]

(a) now 56

Answer

Date 21/02/2021

Program No. 44

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

```
import csv  
csv_columns = ['id', 'column1', 'column2', 'column3']  
dict_data = {'id': [1, 2, 3],  
            'column1': [33, 25, 50]  
            'column2': [35, 30, 30]  
            'column3': [21, 40, 55]}
```

csv_file = "temp.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 (10 errors")

data = csv.DictReader(open(csv_file))

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

for row in data:

 print (row)

Result:

The program has been executed and the output was recorded.

OOP

class constructor

{'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': ...},
...
var equal

(function (req, res) {
 const id = req.params.id;
 const photo = req.query.photo;
 const filter = req.query.filter;
 const sort = req.query.sort;
 const limit = req.query.limit;
 const offset = req.query.offset;
 const count = req.query.count;
 const query = req.query.query;