Area of a circle

Program no: 1 Date: 17/1/2021

Aim: write a python program to return area of a circle using a function

Program:
```
def area(r):
    import math
    a = r**2*math.pi
    return a
r = float(input("enter the radius of the circle:"))
print("%.2f" % area(r))
```

Result: The program has been executed and the output was verified

output:

enter the radius of the circle:3

28.27

Square of a number

program no:.2 Date:17/1/21

Aim: write a python program to find a square of a number entered by the user

program:
```
n=int(input("enter an integer number:"))
square=n*n
print("square of number :",square)
```

Result: The program has been executed and the output was verified

output:

Enter an integer number: 6
Square of number: 36.

Biggest of three numbers

program no:3 Date: 17/12/21

Aim: write a python program to find the biggest of 3 numbers entered by the user

program:

```
first=float(input("enter first number:"))
second = float(input("enter second number"))
third= float(input("enter third number:"))
if (first >second) and (first>third):
        largest=first
elif (second >first) and (second >third):
        largest=second
else:
        largest=third
print("the largest number is", largest)
```

Result: The program has been executed and the output was verified

output

Enter first number: 5

enter second number : 8

Enter third number: 10

the Largest number is : 10.0

Leap year

programno: 4 Date: 27/1/21

Aim: Display feaetur leap years from current year to a final year entered by user.

program:
```
def leap(year):
    c=2021
    while c<=year:
        if c%4!=0 and c%400!=0:
            ct=1
        elif c%100!=0:
            print(c)
            ct=1
n=int(input("Enter the year:"))
leap(2022)
```

Result: the program has been executed and the output was verified

Output:

Enter the year : 2032

2024

2028

2032

Square of N numbers

Program no: 5 Date: 27/1/21

Aim: square of N numbers

program:
```
def square(n):
    sum = 0
    for i in range (1, n+1):
        sum = sum+(i*i)
    return sum
n = int (input("enter a number"))
print (square(n))
```

Result: the program has been executed and output was verified

output

Enter a number : 4

30

Vowels

Program: 6 Date: 27/1/21

Aim: form a list of vowels selected from a given word

program: def check (string):

```
    vowels = "AaEeIiOoUu"
    character = [each for each in
    string if each in vowels]
    print (character)
string = (input (enter a string"))
check (string)
```

Result: the program has been executed and output was verified

output:

Enter a string: "farsana jasmin"

['a', 'a', 'a', 'a', 'i']

word count

Program: 7 Date: 27/1/21

Aim: count the occurrences of each word in a line of text

program:
```
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts
print(word_count("It's okay not to be okay"))
```

Result: The program has been executed and output was verified

output:

{'HS': 1, 'okay': 2, 'not': 1, 'to': 1, 'be': 1}

occurrences of 'a'

program: 8 Date: 27/1/21

Aim: store a list of first names. count the occurences of 'a' within the list

program:

```
def count(string):
    counter=0
    letter='a'
    for x in range(0,len(string)):
        for y in range(0,len(string[x])):
            if letter==string[x][y]:
                counter=counter+1
    print("there are",counter,letter+"'s in the list")
count(string)
```

Result: The program has been executed and output was verified

output:

string = ["amala", "anagha", "anila", "amal"]

there are 10 a's in the list

list length

program: 9

Date 27/1/21

Aim: Enter 2 lists of integers. check whether list are same length

program:

```
def list(l1,l2):
    a=len(l1)
    b= len(l2)
    if a==b:
        print("same length")
    else:
        print ("not same length")

list(l1,l2)
```

Result: the program has been executed and output was verified

output:

l1 = [1, 2, 3, 4, 5]

l2 = [3, 4, 0, 0, 0, 2]

Same length

Sum:

program: 10 Date: 27/1/21

Aim: Enter 2 lists of integers. check whether list sums to same value.

program:
```
def sm(l1,l2):
    a=sum(l1)
    b=sum(l2)
    if a==b:
        print("equal sum")
    else:
        print("unequal sum")

sm(l1,l2)
```

Result: the program has been executed and output was verified

output:

l1 = [1, 2, 3, 4, 5]

l2 = [3, 4, 9, 0, 2]

unequal sum

same occurance of value

program no: 11 Date: 27/1/21

Aim: Enter 2 list of integers. check whether any value occure in both

program:
```
def pn(r):
    if r==1:
        print("there are common elements")
    else:
        print("no common elements")

def cm(l1,l2):
    r=0
    for a in l1:
        for b in l2:
            if a==b:
                r=1
    return r

pn(cm(l1,l2))
```

Result: the program has been executed and output was verified

output:

l1 = [1, 2, 3, 4, 5]

l2 = [3, 4, 7, 9, 0, 2]

there are common elements

Replace

Program no: 12 Date: 27/1/21

Aim: get a string from an input string where all occurances of first character replaced with '$', except first character.

program:

```
def replace(str):
    char = str[0]
    str = str.replace(char,'$')
    str = char+str[1:]
    return str
print(replace('WOW'))
```

Result: The program has been executed and output was verified

output

WO$

program no: 13 Date: 27/1/21

Aim: create a string from given string where first and last character exchanged

program:
```
def exchange(str):
    return str[-1:] + str[1:-1] + str[:1]

print(exchange('hello'))
```

Result: The program has been executed and output was verified

output

oellh

Integer addition

program no: 14 Date: 27/1/21

Aim: Accept an integer n and compute n + nn + nnn

program:
```
def nn(a):
    n1 = a
    n2 = a*a
    n3 = a*a*a
    print(n1+n2+n3)

nn(a)
```

Result: the program has been executed and Output was verified

output

n=4

84

program no: 15 Date: 27/1/21

Aim: sort dictionary in ascending and descending order.

program:
```
def dec(c):
    li = list(c.items())
    li.sort()
    print("ascending order is",li)
    li = list(c.items())
    li.sort(reverse=True)
    print("descending order is",li)

dec(c)
```

Result: the program has been executed and output was verified

output

C= {'apple':4 'orange':3 'mango':2 'banana':1}

nsending order is [('apple',4),('banana',1),

('mango',2),('orange',3)]

descending order is [('orange',3),('mango',2),

('banana',1),('apple',4)]

merge

program no: 16 Date: 27/1/21

Aim: merge two dictionaries

program:
```
def mer(d1,d2)
    res={**d1,**d2}
    return res

print(mer(d1,d2))
```

Result: the program has been executed and output verified

output:

d1 = {'a':1, 'b':2, 'c':3}

d2 = {'e':4, 'f':5, 'g':6}

{'a':1, 'b':2, 'c':3, 'e':4, 'f':5, 'g':6}

gcd

Program no: 17 Date: 27/1/21

Aim: find gcd of two numbers

program: def g(a,b)

```
    import math
    print ("gcd:", math.gcd(a,b))

g(a,b)
```

Result: the program has been executed and output verified

output:

a=60

b=48

gcd=12

Remove

Program: 18 Date: 27/1/21

Aim: from a list of integers, create a list removing even numbers

program:
```
def remove(li):
    for i in li:
        if (i%2==0):
            li.remove(i)
    print("list after removing even numbers:",li)

remove(li)
```

Result: The program has been executed and output verified

output:

li = [21, 22, 23, 24, 25, 26]

list after removing even numbers: [21, 23, 25]

factorial

program no. 19 Date: 27/11/21

Aim: program to find factorial of a number

program:
```
def fact(num):
    factorial = 1
    if num < 0:
        print("factorial does not exist for negative number")
    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)

fact(7)
```

Result: The program has been executed and output verified

output

The factorial of 7 is 5040

Fibonacci series

program no: 20 Date: 3/2/21

Aim: Generate Fibonacci series of N terms

program:
```
def fibo(term):
    n1,n2=0,1
    count=0
    if term<=0:
        print("please enter a positive
            integer")
    elif term==1:
        print("Fibonacci sequence upto",
        term,":")
        print(n1)
    else:
        print("Fibonacci sequence:")
        while count < term:
            print(n1)
            nth=n1+n2
            n1=n2
            n2=nth
            count+=1
fibo(7)
```

Result: program has been executed and output was verified

output:

Fibonacci sequence:

0
1
1
2
3
5
8

sum of lists

program no: 21 date: 3/2/21

Aim: find the sum of all items in a lists

program:

```
def sum_list (items):
    sum_numbers = 0
    for x in items:
        sum_numbers += x
    return sum_numbers.
print (sum_list [3, -4, 6, 7]))
```

Result: The program has been executed and output was verified

output :

12

even & perfect

Program no: 22

Date: 3/2/21

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

Program:

```
def call():
    n=0
    for x in range(1000,10000):
    num= str(x)
    number=int(x)
    first =int (num[0])
    second= int (num[1])
    third = int (num[2])
    fourth= int (num[3])
    if first % 2==0:
        if second %2==0:
            if third %2==0:
                if fourth %2==0:
                    for i in range(2, number):
                    if i*i == number:
```

output:

4624
6084
6400
8464

pyramid

program no: 23 Date: 8/2/21

Aim: display the given pyramid with step number accepted from user

program:
```
def a(lines):
    i=1
    l=1
    while i<=lines:
        i=1
        while l<=i:
            temp=i*1
            print(temp,end="",flush=
            True)
            print(" ",end="",flush=
            True)
            i=i+1;
        print(" ");
        i=i+1;
line=int(input("Enter a number:"))
a(line):
```

result: The program has been executed and output was verified

output

Enter a number: 6

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

number of characters

program no: 24 Date: 3/2/21

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

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('hello world'))
```

Result: The program has been executed and output was verified

output

{'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1,
'r': 1, 'd': 1}

Add 'ing' or 'ly'

program 20: 25 date: 3/2/21

Aim: Add 'ing' at the end of a given string, if it is already ends with 'ing', then add 'ly'

program:

```
def add_string(str1):
    length=len(str1)
    if length>1:
        if str1[-3:]=='ing':
            str1+='ly'
        else:
            str1+='ing'
    return str1
print(add_string('abc'))
```

Result: the program has been executed and output was verified

output

abcing

Length of longest word

program no: 26 Date: 3/2/21

Aim: Accept a list of words and return length of longest word

Program:

```
def find(word):
    wl=[]
    for n in word:
        wl.append((len(n),n))
    wl.sort()
    result=wl[-1][0],wl[-1][1]
    print("longest word:",result[1])
    print("length of the longest word:",
          result[0])

find(["hello","whatsapp","hi"])
```

Result: the program has been executed and output was verified

output:

longest word : whatsapp

length of the longest word : 8

Program no: 27 Date: 3/2/21

Aim: construct following pattern using nested loop

program:
```
def star():
    n=5
    for i in rang(n):
        for i in range(i):
            print("*",end="")
        print("")
    for i in range(n,0,-1):
        for i in rang(i):
            print("*",end="")
        print("")
star()
```

Result: the program has been executed and output was verified

output

factors

Program no: 28. Date: 3/2/21

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)
print_factors(034)
```

Result: the program has been executed and output was verified

output:

The factors of 234 are:

1
2
3
6
9
13
18
26
39
78
117
234

lambda

program no: 29 Date: 3/2/21

Aim: write a lambda functions to find area of square, rectangle and triangle.

program:
```
s=int(input("Enter the length of a side of square:"))
l=int(input("enter the length of rectangle:"))
b=int(input("enter the breadth of rectangle:"))
h=int(input("enter the base of triangle:"))
d=int(input("enter the height of triangle:"))
x=lambda s:s*s
y=lambda l,b:l*b
t=0.5
z=lambda h,d,t,:h*d*t
print("area of square is:",x(s))
```

```
print("area of rectangle:", x(l,b))
print("area of triangle", z(h,d,t))
```

Result: program has been executed and output was verified

output:

Enter the length of a side of square: 4
Enter the length of rectangle: 5
enter the breadth of rectangle: 6
enter the base of triangle: 4
enter the hight of triangle: 5
area of square is: 16
area of rectangle: 30
area of triangle: 10.0

Rectangle class

Program no: 30 Date:

Aim: create Rectangle class 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,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 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():
        print ("Rectangle with length =", obj.
length, " and breadth =", obj.breadth,
"has the greater area")
    else:
        print ("They have equal area!")
r1 = Rectangle(9,3)
r2 = Rectangle(3,4)
r1.cmp(r2)
```

Result: program has been executed and output was verified

output

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

BankAccount

program no: 31 Date:

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

Program:

```
class BankAccount:
    def __init__(self,a,n,t,b):
        self.acno=a
        self.name=n
        self.type=t
        self.bal=b
    def deposit(self,a):
        self.bal=a
        print('Rs.',a,'deposited!
    current balance is:Rs.',self.bal)

    else:
        print('insufficient balance to
        make this transaction!')
```

```
a=int(input("Enter account number:"))
n=input("Enter name of the account
        holder:")
t=input("Enter account type:")
b=float(inpt("Enter your balanace:"))
acl=BankAccount(a,n,t,b)
acl.deposit(float(input("Enter amount
to deposit:")))
acl.withdraw(float(input("Enter amount
to withdraw:")))
```

Result: program has been executed and output was verified

output

Enter account number: 12345678

Enter name of the account holder : farsana

Enter account type = savings

Enter your balance : 12000

Enter amout to deposit : 1000

Rs. 1000.0 deposited! current balance is:

Rs. 13000.0

Enter amount to withdraw 500

Rs: 500. withdrawn! current balance is.

Rs. 125000.

Rectangle class with private

Program no: 32 Date:

Aim: create a class Rectangle with private attributes length and width. overload '<' operator to compare the area of 2 rectangle.

program:

```
class Rectangle:
    def __init__(self,l,w):
        self.__length=l
        self.__width=w
        self.area=self_width* self_breadth
    def __lt__(self,other):
        if self.area<other.area:
            print("Rectangle with length=",
other.__length," and width=",other.__width,
"has the lesser area!")
        else:
            print("They have equal area!")
l=float(input("Enter length of 1st
        rectangle: "))
```

```
w = float(input("Enter width of 1st
        rectangle:"))
R1 = Rectangle(l,w)
l = float(input("Enter length of 2nd
    rectangle:"))
w = float(input("Enter width of 2nd
rectangle:"))
R2 = Rectangle(l,w)
R1 < R2
```

Result: program has been executed and output was verified

output:

Enter length of 1st rectangle: 12

Enter width of 1st rectangle: 5

Enter length of 2nd rectangle: 13

Enter width of 2nd Rectangle: 5

Rectangle with length=12.0 and width=5=0 has the lesser area!

Time class

Program no: 33 Date:

Aim: create a class Time with private attribute hour, minute and second. overload '+' operator to find sum of 2 time

program:

```
class Time:
    def __init__(self, hh=0, mm=0, ss=0):
        self.__hour = hh
        self.__minute = mm
        self.__second = ss
    def __add__(self, other):
        second = int((self.__second + other.__second) % 60)
        minute = int((self.__minute + other.__minute) % 60 + ((self.__second + other.__second) / 60))
        hour = int((self.__hour + other.__hour) % 24 + (self.__minute +
```

```
other.--minute)/60)
print ('Time [hh:mm:ss] ',hour,':'
minute,':',second)

T1 = Time(10, 25, 45)
T2 = Time (16, 45, 56)
T1+T2
```

Result: program has been executed succesfully and output was verified

output

Time [hh: mm: ss] 5:11:41

Publisher

Program no: 35 Date:

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 overloading.

program:
```
class publisher:
    def __init__(self,name1):
        self.name=name1
    def show(self):
        pass
class Book(publisher):
    def __init__(self,title1,
    author1,name1):
```

```
        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 : Rs.", self.price)
```

```
Print ("No of pages :", self.no_of_pages)
P1 = python( 565.90, 250, 'programming with
python', 'GvR Rossum', 'ABC Books')
p1.show()
```

Result: program has been aexecuted success fully and output verified

output

Book : kitty
Author : clara
publisher : cdc books
price : Rs. 200
no of pages : 110

Read file

program no: 36. date:

Aim: write a python program to read a file line by line and store it into a list

program:

```
def file_read (fname):
    with open(fname) as f:
        c = f.readfilelass()
        print(c)

file_read ("file2.txt")
```

Result: program has been executed succesfully and output was verified

odd lines

program no: 37 date:

Aim: pytho program to copy odd lines of one file to other.

program:

```
a=open("file2.txt","r")
b=open("hello2.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("file2.txt","r")
d=b.read()
print(d)
d.close()
b.close()
```

list of strings

program no: Date:

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('ff.csv', newline='') as
csvfile:
d=csv.reader(csvfile, delimeter=
',', quotechar='|')
for r in d:
    print(','.join(r))
```

Result: program has been executed successfully and output was verified

read columns

program no: 39 Date:

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('c1.csv',newline='') as csvfile:
    cl=csv.DictReader(csvfile)
    print("author original_title")
    for r in cl:
        print(r['best_book_id'],r["original_title"])
```

Result: program has been executed successfully and output was verified

output

authors	original title
Suzanne Collins	The Hunger Games
J K Rowling, mary granelpre	Harry potter and Philosophers stone
stephenie meyer	Twilight

Displaey content

program no: 40 Date:

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

program:

```
import cvs
field_names = ['best_book_id', 'authors',
'original_title']
book = [
{
'best_book_id': 1, 'authors': 'suza
nne collins', 'original_title':
The Hunger Games'
}
{
best_book_id': 2, 'authors': 'jok.
```

output

1, suzanne collins, The range names

2, "J.K Rowling, mary grandpre", Harray potter and the philosophers stone

3. stephenie meyer, twilight

```
Rowling, mary grandpre', 'original_title': '
Harry potter and the philosophers
    },
{
'best_book_id': 3, 'authors': 'stephenie
    meyer', 'original_title': 'Twilight'
    },
{
'best_book_id': 3, 'authors': 'stephenie
meyer', 'original_title': 'Twilight'
    },
]
with open('c1.csv','w') as csvfile:
    write=csv.DictWriter(csvfile, fieldnames
=field_names)
    writer.writeheader()
    writer.writerow(book)
with open ('c1.csv', newline='') as csvfile:
```

```
d = csv.reader (csvfile, delimiter=',')
for r in d:
    print (',' .join(r))
```

Result: program has been executed successfully and output was verified

Leap year

program no: 41 Date:

Aim: Display future leap years from current year do a fina year entered by user.

Program:

```
def leap (year):
    c = 2021
    while c<=year:
        if c%4 !=0 and c% 400!=0:
            c+=1
        elif c%100 !=0:
            print(c)
            c+=1
    n= int(input("Enter the year:"))
    leap(n)
```

Result: program has been executed and output was varified

output

enter the year 2030

2024

2028

Positive list of numbers

program no: 42

Date:

Aim: Generate positive list of numbers from a given list of integers

program:

```
list1=[12,-56,7,-8,-9,-5,4,2,1]
pos=list()
for i in list1:
  if i>0:
    pos.append(i)
print("original list:",list1)
print("positive integer list:",pos)
```

Result: program has been executed and output verified

output

original list: [12, -56, 7, -8, -9, -5, 4, 2, 1]

positive integer list: [12, 7, 4, 2, 1]

comma seperated color

program no: 43 Date:

Aim: create a list of colors from comma-seperated color names entered by user. Display first and last colors.

program:
```
colors = (input ("Enter colors
seperated by commas: )) . split
(',') print ("First color:",
colors [0])
print ("Last color :", colors
[len (colors)-1])
```

result: program has been executed and output was verified

output:

Enter colors seperated by commas:
yellow, red, black, green

First color: yellow
Last color: green

color-list

program no: 44 Date:

Aim: print out all colors from color-list 1 not contained in color-list 2

program:
```
colors1 = set (( input (( Enter colors
seperated by commas: )) . split (( , ))
colors2 = set (( input ( ( Enter colors
seperated by commas: )) . split
(( , ))
print (( colors in color list 1 not
contained in color - list 2 are:
1, list (colors1. difference (colors2)))
```

Result: program has been executed and output was verified

output:

Enter colors seperated by commas: red, yellow, blue, black

Enter colors seperated by commas: green, white, orange

colors in colors_list1 not contained in color_list2 are: ['black', 'blue', 'red', 'yellow']

Swapping character

program no:-45 Date:

Aim:- create a single string seperated with space from two strings by swapping the character at position 1

program:
```
Str1 = input ("Enter a string:")
Str2 = input ("Enter another string:")
Str3 = str2[0] + str1[1:] + " " + str1[0]
+ str2[1:]
print (str3)
```

Result:- program has been executed and output verified

output:

Enter a string: hello

Enter another string : world

~~hel~~ wello horld

Package

Program no: 46 Date:

AIM: python program to create a package graphic with modules Rectangle, circle and sub package 3D-graphic with modules cuboid and sphere include methods to find area and perimeter of respective figures in which each module with programs that find area and perimeter of figures by different importing statements

Program:

circle.py

```
def area(r)
print('Area of circle with radius',r,'is:'
 %.2f'%(3.14*r*r),'sq units')
def circumference(r):
print('circumference of circle with
radius',r,'is:',('%.2f'%(3.14*2*r),
'units')
```

Rectangle.py

```
def area(a,b)
  print('area of rectangle with sides;
    'and' 6,'is:'%.0f'%(a*b).
    'squnits')

  def perimeter(a,b):
  Print ( perimeter of regtongle with
      sides', a,'and',6,'is:','%.2.f'
      %.(a*(a+b))-(units)
```

Sphee.py

```
def area():
    print(' area of sphere with radius;
    r,'is:'), '%.2f'%(4*(3.14*r*r))
    sq.units)

def perimeter(r):
    print ('perimeter of (great circle
    of) sphere with radius;r,'is',
    +'%.2f'%.(2*3.14*r)(units')
```

cuboid.py

```
def area (l,b,h):
print (Total surface area of cuboid
with dimensions ;'l',(b',h',(is':,
('%.2f'% (2*((l*b)+(b*h)+(l*h)))
('sq.unit')

def perimeter(l,b,h):
print ('perimeter of cuboid with
dimesion ;',l,',',b,'(',h,'is:',
('%.2f', % (4*(l+b+h)),'units')
```

Find perimeter.py

Impact circle

```
from rectangle import *
from graphics 3D graphics import cuboid
sphere
a=float (input("Enter length of the
rectangle:"))
b=float(input("Enter headth of the
rectangle:"))
```

```
perimeter(a,b)
r= float(input("Enter the radius of the
    circle:"))
circle.circumference(r)
l=float(input("Enter length of the cuboid:"))
b=float(input("Enter breadth of the
  cuboid:"))
  cuboid.perimeter(l,b,h)
  r=float(input("Enter the radius of
the sphere:"))
    shape.perimeter(r)

findarea.py
  import circle
From rectangle import*
From graphics_3D graphics import cuboid
sphee
a=float(input("Enter length of the
    rectangle:"))
b=float(input("Enter breadth of the
```

```
area(a,b)
l=float(input("Enter length of the
    cuboid:"))
b=float(input("Enter breadth of the
    cuboid:"))
h=float(input("Enter height of the
    cuboid:"))
cuboid.area(l,b,h)
r=float(input("Enter radius of the
sphere:")
shere.area(r)
```

Result: program has been executed and output was verified

output

Enter length of the rectangle: 4
enter breadth of the rectangle: 3
Perimeter of rectangle with sides 4.0 and 3.0 is: 14.00 units

Enter the radius of the circle: 2
Circumference of circle with radius 2.0 is: 12.56 units

Enter length of the cuboid: 5
Enter the breadth of the cuboid: 4
Enter the height of cuboid: 3
perimeter of the cuboid with dimension so 4.0, 3.0 is 48.00 units.

Enter the radius of the sphere: 2
Perimeter of (great of circle of) sphere with radius 2.0 is 12.56 units

Enter length of the rectangle: 2
Enter breadth of the rectangle: 3
Area of rectangle with side 2.0 and 3.0 is: 6.00 sq. units

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: 7
Enter height of the cuboid: 2
total surface area of cuboid with dimension 4.0, 7.0, 2.0 is 100.00 sq. units

Enter the radius of the sphere: 1
area of sphere with radius 1.0 is 12.56 sq. units.