Program No: 28

Aim: display future leap years from current year to final year entered by user

```
import datetime
a = datetime.datetime.now()
a = int(a.year)
b = int(input("Enter final year:"))
print("The leap years:")
for i in range(a, b+1):
if(i%4==0):
print(i)
```

Result: The program has been executed and output is verified

OUTPUT

enter final year 2021

leap year : 2020.

Program No: 29

Aim: List comprehensions

(a) generate positive list of numbers from a given list of integers

```
list1 = [1, 81, 9, -5, 9, -2, -64, 87, -33, -76, 94, -67]
pos = list()
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 output is verified.

OUTPUT

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

positive integer: [1, 2, 9, 87, 24]

Program No: 80

Aim: 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 b>c:
print(a, "is the biggest number")
elif b>a and b>c:
print(b, "is the biggest number")
else:
print(c, "is the biggest number")
```

Result: The program has been executed and output is verified

OUTPUT

enter 1st no: 2

enter 2nd no: 3

enter 3rd no: 4

4 is the biggest number.

Program NO: 31

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

colors = input('enter colors seperated by commas:')).split(',

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

print('Last color:', colors[len(colors)-1])

Result: the program has been executed and output is verified

OUTPUT

enter colors seperated by commas : blue

first color: blue

last color: blue.

Program No: 89

Aim: Print out all colors from color-list1 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 output is verified

OUTPUT

enter colors seperated by commas: red

enter colors seperated by commas: blue

colors in color list1 but contained in color list

are: ['red']

Program No: 33

Aim: Create a single string seperated with space from two strings by swapping the characters at position 1.

```
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 is received

OUTPUT

enter a string: amma

enter another string: appa

amma appa

62.83185307179586

4.15926535897932

f width 10 is: 100

b and width 10 is: 40

ght 10 is: 600

th, height 10 is: 120

56.637061435917 8

10 is 62.83185907179

Program no: 34

Aim: create a package graphics with modules rectangle, circle and sub package 3D-graphics with modules cuboid, and sphere include methods to find area and perimeter of respective figures in each module. write programs that finds area and perimeter of figures by different importing statements (include selective import of modules and import * statements)

circle

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

area

```
import circle
from rectangle import *
from graphics.3D-graphics import cuboid, sphere
a=float(input('enter length of the rectangle:'))
b=float(input('enter breadth of the rectangle:'))
area(a,b)
r=float(input('enter the radius of the circle:'))
circle.area(r)
l=float(input('enter length of the cuboid:'))
b=float(input('enter breadth of the cuboid:'))
h=float(input('enter height of the cuboid:'))
```

Output

Perimeter of circle with radius 10 is 62.83185307179586

Area of a circle with radius 10 is: 314.1592653589793

Area of a Rectangle with length and width 10 is: 100

Perimeter of a Rectangle with length and width 10 is: 40

Area of cuboid with length, width, height 10 is: 600

Perimeter of cuboid with length, width, height 10 is: 120

Area of sphere with radius 10 is: 1256.6370614359178

Perimeter of sphere with radius 10 is 62.83185307179

9586.

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

perimeter

```
import circle
from rectangle import *
from graphics._3D_graphics import cuboid,sphere
a=float(input("Enter length of the rectangle:"))
b=float(input("Enter breadth 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:"))
h=float(input("Enter height of the cuboid:"))
cuboid.perimeter(l,b,h)
r=float(input("Enter the radius of the sphere:"))
sphere.perimeter(r)
import circle
rectangle
```

```
def area(a,b):
print("Area of rectangle with sides",a,"and",b,"is:",
      1/2*(a*b),"sq.units")
def perimeter(a,b):
print("perimeter of rectangle with sides",a,"and",
      b,"is:",1/2*(2*(a+b)),"units")
3d graphics
```

cuboid

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

sphere

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

Result: The program has been executed and output is verified.

= 3 has the

Program NO: 35

Aim: Create Rectangle class with attributes length and bredth and methods to find area and perimeter. Compare two Rectangle objects 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 and breadth =', obj.breadth, 'has greater
area')
else:
print('they have equal area!')
r1=Rectangle(9,3)
r2=Rectangle(3,4)
r1.comp(r2)
```

Result: The program has been executed and output is verified

OUTPUT

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

Program NO:36

Aim: create a bank account members account number, name, type of account and balance. use the constructor and methods to deposit at the bank and withdraw an account from the bank.

```
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! Your updated balance is: Rs.',
      self.bal)
def withdraw(self,a):
 if self.bal>=a:
 self.bal-=a
print('Rs.',a,'withdrawn! 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:')
```

OUTPUT

enter account number : 10203015

enter name of the account holder : anju thomas

enter account type : saving

enter your balance : 100006

enter amount to deposit 1000

RS. 1000.0 deposited! current balance is : RS. 101000.0

enter amount to withdraw : 1000

RS. 1000.0 withdrawn! current balance is : 100000.0

```
t= input ('enter account type:')
b= float (input ('enter your balance :'))
ac1 = Bankaccount (a,n,t,b)
ac1. deposit (float (input ('enter amount to deposit:')))
ac1. withdraw (float (input ('enter amount to withdraw:')))
```

Result: The program has been executed and output is verified

e: 10
gle: 10
gle: 5
tangle: 5
d width = 5.0

Program NO: 37

Aim: create a class Rectangle with private attributes length and width width. overload <.

```
class Rectangle:
def __init__(self,l,w):
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.are:
print('Rectangle with length =', other._length, and
width=', other._width 'has 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: The program has been executed and output is verified

Output

enter length of 1st rectangle: 10

enter ~~length~~ width of 1st rectangle: 10

enter length of 2nd Rectangle: 5

enter width of 2nd Rectangle: 5

Rectangle with length = 5.0 and width = 5.0 has the lesser area!

Program NO: 88

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

Class time:

```
def __init__ (self, 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+int ((self._
second +other._second)/60))
hour =int ((self._hour +other._hour )%24 +(self._minute
+ other._minute)/60)
print ("time [hh:mm:ss]", hour, ":", second)
t1= time (12, 45,45)
t2 = time (16,45,56)
t1+t2
```

Result: The program has been executed and output is verified.

Output

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

Program No: 89

Aim: create a class published(name). Derive class Book from published and attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Create a program to display information about a python book. Also have demonstration of multiple and method overriding

```
class published:
def __init__(self,name1):
self.name = name1
def show(self):
  pass
class Book(published):
def __init__(self,title1,author1,name1):
self.title = title1
self.author = author1
published.__init__(self,name1)
def show(self):
pass
 class python(Book):
def __init__(self,price,title1,author1,name1):
```

OUTPUT

Book title : programming with python

Author : GV Rossum

Published : ABC BOOKS

Price : RS. 565.9

No of pages : 250

```
        self.price=p
        self.no_of_pages=no_Book:_init_(self,title1,author1,name1):
    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_page)
p1=python(565.90,250,'Programming with Python',
        'GV Rossum','BBC Books')
```

Result: the program has been executed and output is verified.

Program No: 40

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

```
def file_read(fname):
    with open(fname) as f:
        c = f.readlines()
    print(c)
    # Print(len(c))
file_read("file2.txt")
```

Result: The program has been executed and output verified

Output

['0. print("hello")\n', '1. hai\n', '2. file handling\n',
'3. sggshcgbjcvb bcbghdgchjvn dgcb sydbcvfbg
1kjhgydfik dgcvbyfkidoy sgvbcvf kcsxbx cv fhcgb
ncmd jhcn dbhfdj,m']

python program no: 41

Aim: python program to copy odd lines of one file to another

```
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)
a.close()
b.close()
```

Result: The program has been executed and output is verified

Output

1. hai
3. sggshc ghjcvb hcbghdgchjvn.

Program No: 49.

Aim: write a program to read each row from a given csv files and print as list of strings?

```
import csv
with open('cs.csv', newline='') as csvfile:
d = csv.reader(csvfile, delimiter=',', quotechar='|')
for row in d:
print(', '.join(row))
```

Result: The program has been executed and output is verified

Output

hai, hello

xdefghvb, jklk jhydfdrstxc, vbnmhgfv,, xm,, nxm,,

csgrghjdkji

ghnj,, dmkjhgfdszxcvx, bncl,ggr, dxcv xbnn, vcv cnb xba

Program no: 48

Aim: write a program to read specific columns of given csv files and print the content of the columns

```
import csv
with open('c1.csv', newline='') as csvfile:
d= csv.DictReader(csvfile)
print("authors original_title")
for a in d:
print(a['authors'],a['original_title'])
```

Result: The program has been executed and output is verified.

output

author	original_title
suzanne collins	the Hunger games
J.K rowling, mary grand pre	Harry potter and the philosphers stone
Stephenie meyer	twilight.

potter and the

Program no: 4.4

Aim: write a program to write a Python dictionary to a csv file. Later write the csv file. Read the csv file and display the content

```
import csv
field_names = ['best_book_id', 'authors', 'original_title']
book = [
  {
'best_book_id':1,'authors':'suzzanne collins', 'original_title':'
}, the hunger games'
{
'best_book_id':2,authors':' JK. Rowling, mary GrandPre',
'original_title':' Harry potter and the Philosphers stone'
},
{
'best_book_id': 3,'authors':' stephenie meyer', 'original_
    title': 'twilight'
  },
]
with open ('cl.csv', 'w') as csvfile:
writer = csv.DictWriter (csvfile, fieldnames = field_names)
writer.writeheader ()
writer.writerows (book)
```

```
with open ('c1.csv', newline ='') as csvfile:
d= csv.reader (csvfile, delimiter = ' ')
for a in d:
print (','.join (a))
```

Result: The program has been executed and output is verified

Output

book_id, authors, original_title.

1, suzanne collins, The Hunger Games

2, "J. K. Rowling, Mary GrandPre', Harry Potter and the Philosophers stone

3, Stephenie Meyer, Twilight.