Program No - 27

Aim: From a list of integers, create a list removing even number?

Program

```
list = [11, 22, 33, 44, 55, 66]
print ("Original list")
print (list)
for i in list:
    if (i % 2 == 0):
        list.remove (i)
print ("list after removing an even number:")
print (list)
```

Result

The program has been executed and Output was verified.

Output

original list

[11, 22, 33, 44, 55, 66]

list after removing an even numbers:

[11, 33, 44, 55, 66]

list after removing an even numbers:

[11, 33, 55, 66]

list after removing an even numbers:

[11, 33, 55]

Program No - 28

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

program

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

Result: The program has been executed and output Verified.

Output

Enter final year : 2050

Leap years :

2024

2028

2032

2036

2040

2044

2048

Program No-29

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

Program

```
list1 = [5, -8, 69, 57, -55, 24, -66, -20, 80, -33, -639, 852]
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 Verified.

Output

Original list : [5, -8, 69, 57, -55, 24, -66, -20, 80, -33, -639, 852]

Positive integer list : [5, 69, 57, 24, 80, 852]

Program No-30

Aim : Find biggest of 3 numbers entered

Program

```
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 verified

Output

Enter 1st no : 562

Enter 2nd no : 960

Enter 3rd no : 750

960 is the biggest number.

Program No-31

Aim: Create a list of colors from comma-separated 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

The program has been executed and hence verified

Outpat

Enter colors seperated by commas : blue, green, red, black, gellow, ligbtgreen

First color : blue

last color : ligbtgreen

Program-No-32

Aim: Paint out all colors from color-list1 not contained in color-list 2.

Program

```
colors1 = set((input('Enter colors seperated by commas:')).split
                (',',))
colors 2 = set((input('Enter colors seperated by commas:')).
                Split(',',))
Print('Colors in color-list1 not contained in color-list 2 are:', list
      (colors1.difference(colors 2)))
```

Result

The program has been executed and Output is verified.

Output

Enter colors seperated by Commas : red, orange, green, black

Enter colors seperated by Commas : blue, yellow, gold, brocoo,
evergreen, lightblue.

colors in color-list 1 not contained in color-list 2 are: ['red', 'orange',
'green', 'black']

Program No-33

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

The program has been executed and output is verified.

Output

Enter a string : avani pa

Enter another string : palikkambanmalkalil

Pvani p a palikkambanmalkalil

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)

Programs.

Circle.py

findarea.py

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

findperimeter.py

```
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:'))
b = float(input('Enter height of the cuboid:'))
cuboid.perimeter(l,b,b)
r = float(input('Enter the radius of the sphere:'))
sphere.perimeter(r)
```

rectangle.py

```
def area(a,b):
    print('Area of rectangle with sides',a,'and',b,'is:','%.2f'%
        (a*b),'sq.units')
def perimeter(a,b):
    print('perimeter of rectangle with sides',a,'and',b,'is:',
        '%.2f'%(2*(a+b)),'units')
```

3D-graphics

cuboid.py

```
def area(l,b,b):
```

```
print('Total surface area of cabad with dimensions',l,',',b,',',b,
is:',"%.2F" %(2*((l*b)+(b*b)+(l*b))),'sqcubits')

def perimeter(l,b,b):
  print('perimeter of cuboid with dimensions',l,',',b,',',b,
'is:',"%.2F"%(4*(l+b+b)),'cubits')
```

Sphere.py

```
def area(r):
    print('Area of sphere with radius',r,'is:',"%.2F"%
    (3.14*r*r)),'sq.cubits')

def perimeter(r):
    print('perimeter of (great Circle of) sphere with radius',r,'is:
',"%.2f"%(2*3.14*r),'cubits')
```

Result

The program has been executed and output is verified.

```
print('Total surface area of cabad with dimensions',l,',',b,',',b
is:',"%.2F" %(2*((l*b)+(b*b)+(l*b))),'sqcubits')

def perimeter(l,b,b):
  print('perimeter of cuboid with dimensions',l,',',b,',',b,
'is:',"%.2F"%(4*(l+b+b)),'cubits')
```

Sphere.py

```
def area(r):
    print('Area of sphere with radius',r,'is:',"%.2F"%
    (3.14*r*r)),'sq.cubits')

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

Result

The program has been executed and output is verified.

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 :1256.6370614359173
Perimeter of sphere with radius 10 is 62.83185307179586.

Program No - 35

Aim: Create Rectangle class with attributes length and breadth. and methods to find area and peameter. 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 peaimeter(self):
        return 2* (self.length + self.breadth)
    def cmp(self, obj):
        if self.area() > obj.area():
            print('Rectangle with length=', self.length,
            'and breadth,' has the greater area')
        elif self.area()> obj.area():
            print ('Rectangle with length=', obj.length,)
        and breadth =', obj.breadth, 'has the greater are')
    else:
```

print("They have equal area")

r1 = Rectangle(6,5)

r2 = Rectangle(8,4)

r1.crop(r2)

Result

The program has been ~~Compl~~ Executed and Output is verified.

Output

Rectangle with length = 8 and breadth = 4 has the greater area.

Program No-36

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)

    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:")
t = input("Enter Account type:")
b = float(input("Enter your balance:"))
acl = BankAccount(a,n,t,b)
acl.deposit(float(input("Enter amount to deposit:")))
acl.withdraw(float(input("Enter amount to withdraw:"))
```

Result

The program has be executed and output is verified.

Output

Enter account number: 120052369865

Enter name of the account holder: avani p a

Enter account type: zero balanced account

Enter your balance : 1052

Enter amount to deposit : 500

Rs. 500.0 deposited, Current balance is Rs: 1552.0

Enter amount to withdraw : 200

Rs. 200.0 withdrawn, Current balance is Rs : 1352.0

Program No-37

Aim: Create a class Rectangle with private attributes length and width. Overload 'c' 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.__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 = 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 : 5

Enter width of 1st rectangle : 3

Enter length of 2nd rectangle : 9

Enter width of 2nd rectangle : 6

Rectangle with length = 5.0 and width = 3.0 has the lesser area.

Program No-38

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.__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(2, 05, 45)
T2 = Time(18, 50, 45)
T1 + T2
```

Result:

The program has been executed and output is verified

Output

Time [hh:mm:ss] 21 : 16 : 30

Program No-39

Aim: Create a 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,name1):
        self.name = name1
    def show(self):
        pass
class Book(Publisher):
    def __init__(self,title1,author1,name1):
        self.title = title1
        self.author = author1
        Publisher.__init__(self,name1)
    def show(self):
        pass
class Python(Book):
    def __init__(self,p,no,title1,author1,name1):
        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-pages)

P1 = Python(423.50, 302, 'An Idealist View of Life', 'Dr. S.
    Radhakrishnan', 'Ardeslte Press')

P1.show()
```

Result

The program has been executed and Output is verified.

Output

Book Title : An Idealist View of Life

Author : Dr S. Radhakrishnan

Publisher : Andesite Press

Price : Rs. 423.5

No. of pages 302.

Program No-40

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.readlines()
        print(c)
file_read("file2.txt")
```

Result

The program has been executed and Output is verified.

Output

["1. I'm not going to go deep into history matter.\n\n 2. This introduction was just to make it a bit fun and show how the phrases give us an indication of the importances of pictures!]

Program No - 41

Aim: Python program to copy odd lines of one file to another.

Program

```
a=open ('file0.txt','w')
b=open ('hello0.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. I'm not going to go deep into historical matter.

2. This introduction was just to make it a bit fun, and show the phrases give us an indication of the importance of picture.

Program No-42.

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

Program

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

Result

The program has been executed and output is verified

Output

mert, vaolable	filled	jobs
BDCQ.SEE1045A	2015.06	17596.
BDCQ.SEE1045A	2015.09	17565
BDCQ.SEE1045A	2015.12	17955
BDCQ.SEE1045A	2016.03	17768

Program No-43

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

Program

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

Result

The program has been executed and output is verified

Output

authors	original_title
Suzanne Collins	The Hanger Games
J.K. Rowling, Mary GrandPre	Harry Potter and the philosophers stone
Stephenie Meyer	Twilight

Program No-44

Aim: Write a program to write a python dictory to csv file. After writing the csv file read the csv file and display the contents.

Program

```
import csv
field_names = ['best-book-id', 'author', 'orginal-title']

book = [
    {
'best-book-id': 1, 'author': 'Suzanne Collins', 'Original-title':
    'The Hunger Games'
},
{
'best-book-id': 2, 'author': 'J.K. Rowling, Mary GrandPre',
    'Original-title': 'Harry Potter and the philosopher stone'

},
{
'best-book-id': 3, 'author': Stephenie Meier', 'original-title':
    Twilight'

},
]
with open('ci.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field-names)
```

```
cowriter.writeheadl()
writer.writerows(book)

with open('c1.csv', newline = '') as csvfile:
    d = csv.reader(csvfile, delimeter = '|')
    for rind d:
        print(','.join(r))
```

Result

The program has been executed successfully and output is verified.

Output

best-book-id	authors,	original-title
1	Suzanne Collins,	The Hunger Games
2	J.K Rolling, Mary GrandPre,	Harry Potter and the Philosopher's stone
3	Stephenie Meyer	Twilight.