28/2/2020

Program : 1

Aim :- Python program to Find area of a circle ($A = \pi r^2$) [using Function]

Procedure :-

```
# Area of a circle
From math import pi
r = float (input ("Enter the radious of the circle :"))
Print ("the area of the circle with radious" +
        str (r) + " is : " + str (pi * r * r))
```

Result : the program has been executed and the output was varified.

Output

```
Enter the radious of the circle : 4
the area of the circle radious 4 is 50.2654
```

26/2/21

Program : 2

Aim: Python program to Find largest among three numbers.

Procedure: -

```
# Largest among three numbers.
num1 = float (input ("Enter First number"))
num2 = float (input ("Enter Second number"))
num3 = float (input ("Enter three number"))
if (num1 > num2) and (num1 > num3): largest = num1
elif (num2 > num1) and (num2 > num3): largest = num2.
else: largest (num1 = num3)
print ("the largest number is", largest)
```

Result: the program has been executed and the output was varified.

Output

```
Enter First number : 33
Enter Second number : 25
Enter Third number : 24
the largest number is 33.0
```

28/2/24

Program - 3

Aim:- Python program to find Square of a number

Procedure:

```
# Square of a number
for x in range(100):
print(x**2)
digit = int(input("Enter an integer number:"))
Square = digit * digit
print(f'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
```

Program: 4

Aim :- Python program to find square of n number in list.

```
[   list - [12, 13, 14, 15]
    for n in list:
        print (n*n)   ]
```

Procedure :-

```
# Square of n number in list
list 1 = [12, 13, 14, 15]
for n in list1:
    Square = n*n
print (n, Square is :" Square)
```

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

Output

```
12 Squre is : 144
13 Squre is : 169
14 Squre is : 196
15 Squre is : 225
```

Program : 5

Aim: Python program to Find vowels in a string

Procedure:

```
# vowels in a string

String A = " Hello ... how are you"
Print ("given string \n", stringA)
 Vowels = "AaEeIiOoUu"
res = set ([each for each in string A if each
                in vowels])
Print ("the vowels present in the string:\n",res)
```

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

Output

Given String:

Hellow... how are you

The vowels present in the string:

{'a', 'o', 'o', 'e'}

Program: 6

Aim: Python program to count words in a sentence

Procedure :-

```
# count words in a sentence
original_string = "Python is an interpreted high_level
programming and general_purpose programming
language"
print("The original string is: "+ original_string)
result = len(original_string.split())
print("The number of words: "+ str(result))
```

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

Output

Orginal string = 'python is an interpreted high_level programming and general_purpose programming language'

the number of words : 12

Program: 7

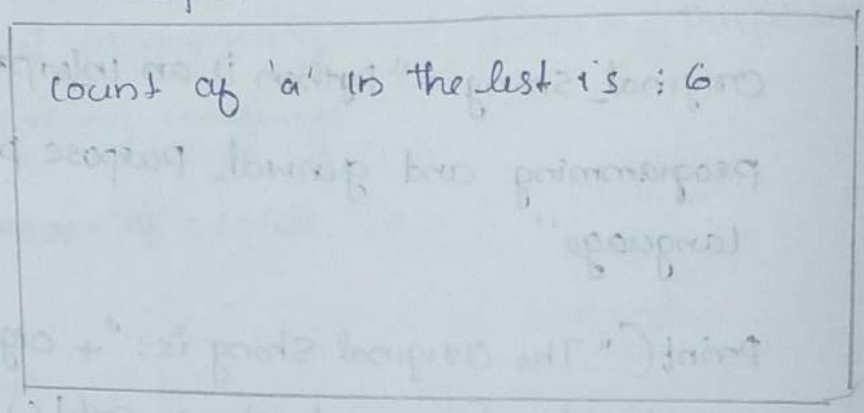
Aim:- Python program to count a letter 'a' in a list.

Procedure:-

```
# count a letter a in a list

a = ['Aiswariya', 'Amala', 'Sujith', 'Sumith']
Str1 = (''.join(a))
count = 0
for i in Str1:
    if i == 'a':
        count = count + 1
print ("count of 'a' in the list is:" str(count))
```

Output



Program : 8

Aim :- Python program to check the length of a list.

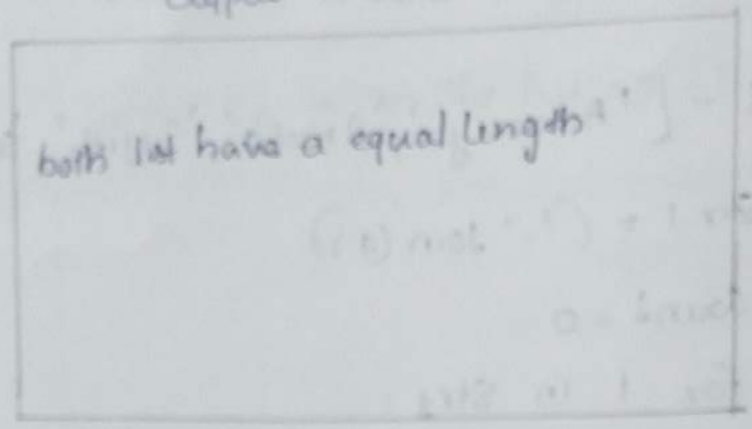
Procedure :

```
# check the length of a list
list1 = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
list 2 = [140, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210]

len1 = len(list1)
len2 = len(list2)
if len1 == len2:
print ('both list have a equal length')
else:
print('both list does not have equal length')
```

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

output



Program: 9

Aim:- Python program to check the sum of list

Procedure:-

```
# Sum of list

total = 0
List1 = [11, 5, 17, 18, 23, 05, 70, 31]
for ele in range(0, len(list1)):
    total = total + list1[ele]
print("Sum of all elements in given list:", total)
```

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

Output

Sum of all element in given list : 200

Program: 10

Aim :- Python program to check the common elements in the list

Procedure :-

```
# common elements in the list
list 1 = [12, 16, 15, 16, 14, 13, 12, 12, 11, 10, 10]
list 2 = [12, 10, 12, 11, 10, 16, 15, 14, 13, 12, 16]
For value in list 1:
    if value in list 2
        common = 1
if common == 1:
print ('there are common elements')
else:
Print ('no common elements')
```

Output

```
there are common elements
```

Program: 11

Aim: Python program to replace a character in a string.

Procedure:-

```
# Python program to replace a character in a string

str1 = input("Please Enter your own string:")
ch = input("Please enter your own character:")
newch = input("Please Enter the new character:")

str2 = ""
for i in str1:
    if(i == ch):
        str2 = str2 + newch
    else:
        str2 = str2 + i
print("\n original string:", str1)
print("modified string:", str2)
```

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

Output

```
Please Enter your own string : Swathy
Please Enter your own character : y
Please Enter your own the new character : i

original string : Swathy
Modified string : Swathi
```

2/3/21

Program: 12

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

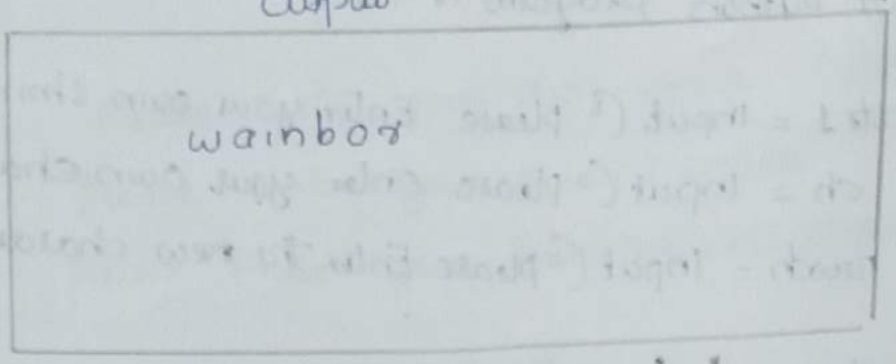
Procedure :-

```
# Exchange the first and last letters in a string

def exchange (Str)
return Str[-1] + Str[1:-1] + Str[:1]
Print (exchange ('rainbow'))
```

Result: The program has been executed and output was verified

output



Program: 13

Aim :- Python program to merge two dictionaries

Procedure :-

```
# merge two dictionaries
d1 = {'a': 100, 'b': 200}
d2 = {'c': 300, 'd': 400}
dictionary = d1.copy()
dictionary.update(d2)
print("the merged dictionary is:", dictionary)
```

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

Output

the merged dictionary is {'a':100, 'b':200, 'c':300, 'd':400}

Program: 14

Aim:- python program to ascend and descent dictionary.

procedure:-

```
# 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("Dictionary in ascending order by value :",
                                    sorted_d)
```

Result: The program has been executed and output was verified

Output

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

Dictionary in asending order by value :

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

Dictionary in descending order by value :

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

9/3/21

Program: 15

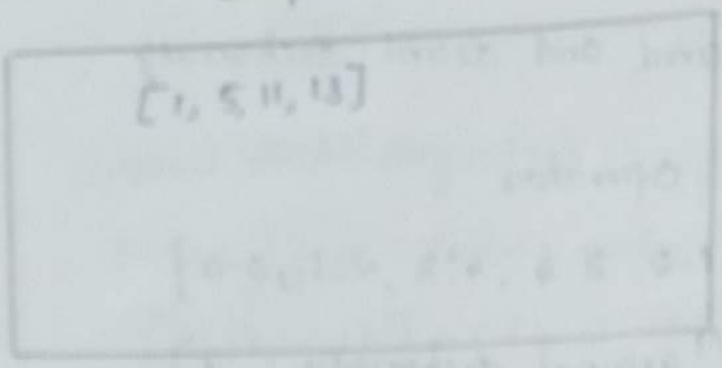
Aim:- Python program to ~~Find~~ remove even numbers from a list.

Procedure:-

```
# remove even numbers from a list
List = [1, 2, 4, 5, 6, 8, 10, 11, 13]
for i in list.copy():
divid = i % 2
if divid = 0:
 list.remove(i)
 print(list)
```

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

output



4/3/21

Program: 16

Aim :- Python program to find gcd of a number.

Procedure :-

```
# Find gcd of a number
num1 = int (input ("Enter a 1st number:"))
num2 = int (input ("Enter the 2nd number:"))
i=1
while 1<=num and 1<= num2):
if (num1 %1 == 0 and num2 %1 == 0):
gcd = i
i = i+1
print ("gcd is", gcd)
```

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

Output

```
Enter a first number : 5
Enter the 2nd number : 15
        gcd is 5
```

4/3/21

Program: 17

Aim:- Python Program to Find Factorial of a number.

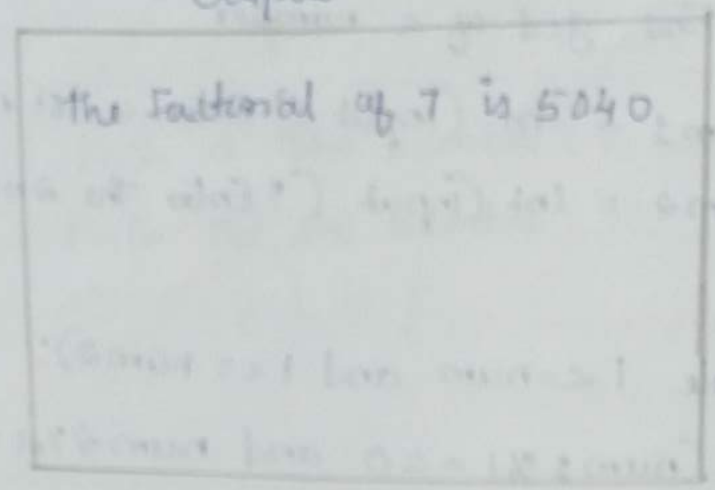
Procedure :-

```
# Find factorial of a number
# Factorial number provided by a user

num = 7  # change value for different result
#num = int (input ("Enter a number :"))
factorial = 1
if num < 0:
print(" Sorry , factorial does not exist for 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 output was verified.

Output



9/8/21

Program - 18

Aim:- Python program to find Fibonacci series.

Procedure:-

```
# Python program to generate Fibonacci series.

n = int(input("Enter the value of n:"))
a = 0
b = 1
Sum = 0
count = 1
print("Fibonacci Series:", end = " ")
while(count <= n):
print(sum, end = " ")
count += 1
a = b
b = sum
sum = a+b
```

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

Output

Enter the value of 'n': 5

Fibonacci Series: 0 1 1 2 3

4/8/21

Program: 19

Aim:- Python program to perform string functions.

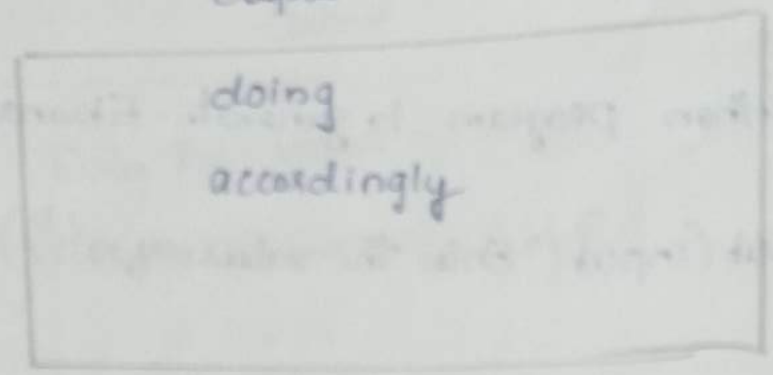
Procedure:-

```
# String Functions

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

    print (add_string ('do'))
    print (add_string ('according'))
```

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


output
doing
accordingly

Program - 20

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

procedure :-

```
for i in range (1000, 10000, 1);
for j in range (32, 100, 1);
    if i == j*j:
        String = str(i)
if string:
    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 excuted. the output was verified.

Output

```
4624
6084
6400
8464
```

Program no : 21

Aim :- Display the given pyramid with step number accepted from user.

Procedure :-

```
rows = int (input("enter the number of rows:"))
for i in range (1, rows):
    for j in range (i, i+1):
    print (i*j, end = ' ')
    print ()
```

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

Output

```
Enter the number of rows: 6
1
2 4
3 6 9
4 8 12 16
```

Program : 22

Aim :- Count the number of characters (Character Frequency) in a string.

Procedure :-

```
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'))
```

output

{'f':1, 'a':1, 'c':2, 'e':1, 'b':1, 'o':3, 'k':1, '.':1, 'm':1}

Program: 23

Aim:- Add 'ing' at the end of a given string. If it already end with 'ing' then add 'ly'

Procedure:

```
def add_string(str1):
    length = len(str1)
    if length > 2:
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else
            str1 += 'ing'
    return str1
print(add_string('crying'))
print(add_string('think'))
print(add_string('string'))
```

output:

```
crygingly
thingking
shingly
```

Program : 24

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

Procedure :-

```
def longestLength(a):
    max1 = len(a[0])
    temp = a[0]
    for i in a:
        if (len(i) > max1):
            max1 = len(i)
            temp = i
    print("the word with the longest length is:"
          temp," and length is", max1)
a = input("Enter a list elements seperated by space")
a = a.split()
longestLength(a)
```

Output

Enter a list elements seperated by space India is my country
the word with the longest is: country and length is 7

Program: 25

Aim:- Construct following pattern using nested loop.

Procedure:

```
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 (" ")

stop:)
```

Result

The program has been executed and the output was verified

Output

The factors of 232 are

1
2
4
8
29
58
116
232

Output

Enter the number of rows: 9

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Program: 26

Aim:- Generate all factors of a number

Procedure:-

```
def print_factors(x):
print("The factors of", x, "are:")
for i in range(1, x+1):
if x%i ==0:
print(i)
num = 10
print_factors(num)
```

output

```
The factors of 10 are:
1
2
5
10
```

Program: 27

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

Procedure:

```
import math.
s_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", s_area (10, 20))
print ("area of circle (15) is : " c_area (20))
print ("area of triangle (12, 20) is", t_area (24, 20))
```

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

Output

```
area of Rectangle (30, 20) is 200
area of circle (15) is : 1256.6370614359173
area of triangle (12, 20) is 220.0
```

Program: 28

Aim: work with build-in packages.

Procedure:

```
import platform
x = platform.system()
Print (x)
print ()
x = dir (platform)
Print (x)
Print ()
import datime
x = datime.datime.now()
Print(x)
x = datime.datime.now()
Print (x.year)
Print (x.strftime ("%A"))
```

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

output

```
2021-03-28  04:18:38.347291
2021
Sunday
```

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.

Procedure:-

```
# class rect1:
# count = 0
# def __init__(self, b, l):
# self.b = b
# self.l = l
# rect1.count = rect1.count + 1
# def area(self):
# return self.b * self.l
# def peri(self):
# return 2 * (self.b + self.l)
# l1 = input("Enter the length:")
# b1 = input("enter the breadth:")
# l2 = input("enter the length:")
# b2 = input("enter the breadth:")
# a1 = rect(l1, b1)
# a2 = rect(l2, b2)
# if a.area == c.area:
# print("equal")
# else:
# print("not equal")
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 rectangle: "))
b = int(input("Enter breadth of rectangle: "))
c = int(input("Enter length of rectangle: "))
d = int(input("Enter length of rectangle: "))
obj = Rectangle(a,b)
obj1 = Rectangle(c,d)
print("Area of 1st rectangle:", obj.area())
print("Area of 2nd rectangle:", obj1.area())
print("Perimeter of 1st rectangle is:", obj.peri())
print("perimeter of 2nd rectangle is:", obj1.peri())
if obj.area() == obj1.area():
print("Equal")
else
print("not equal")
```

Result: the program has been executed. the output ~~of~~ the verified.

Output

```
Enter length of 1st rectangle : 5
Enter breadth of 1st Rectangle : 10
Enter length of 2nd Rectangle : 7
Enter breadth of 2nd Rectangle : 10
Area of first Rectangle : 50
Area of 2nd rectangle : 70
Perimeter of 1st Rectangle : 30
Perimeter of 2nd rectangle : 34
not equal.
```

Program: 30

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

Procedure :-

```
# class Bank():
# def __init__(self):
# self.balance = 0
# print("Your Account is Now Created")
# def deposit(self):
# amount = float(input("\n enter the amount to be
                deposited: "))
# self.balance += amount
# print("\n your amount is deposited :", amount)
# def withdraw(self):
# amount = float(input("\n enter the amount
                to be withdrawn:"))
# if self.balance >= amount:
    self.balance = amount
    print("\n Your withdrew :", amount)
  else:
    print("\n Insufficient balance")
  def display(self):
    print("\n Net available Balance =", self.balance)
  s = Bank_Account()
  s.deposit()
  s.withdraw()
  s.display()
```

Output

```
Enter amount to be deposited : 900
    Amount Deposited  : 900.0
Enter amount to be withdrawn :400
    You withdrew : 400.0
Net available balance : 500.0
```

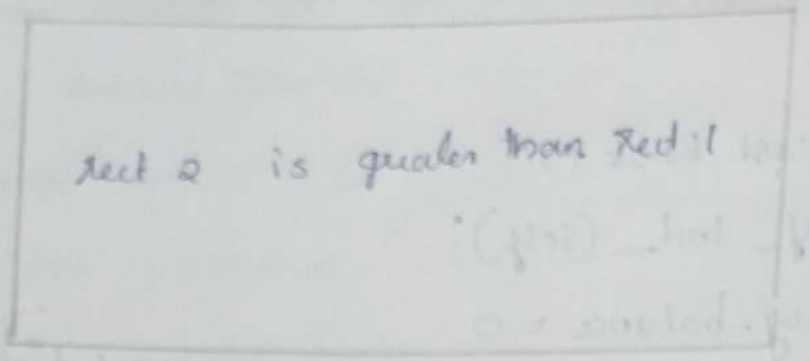
Program: 81

Aim: Create a class Rectangle with private attributes length and width overload '<' to compare the area of two rectangles

Procedure:

```
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
rect 1 = A(3,4)
rect 1. area()
rect 2. A (6,5)
rect 2. area()
if (rect 1 > rect 2):
print ("rect 1 is greate than rect 2")
else:
print ("rect 2 is greater than rect 1")
```

output



Program : 32

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

Procedure :-

```
class Time:
def __init__(self, hour, minut, second):
 self._hour = hour
 self._minut = minut
 self._second = second

 def __add__(self, other):
 h = self._hour + other._minut
 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
```

```
        h = h-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 (':'))
h2, m2, s2 = map (int, time2.split (':'))
t1 = Time (h1, m1, s1)
t2 = Time (h2, m2, s2)
t1+t2.
```

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

Output:

Enter First time in the Format HH:MM:SS

0:2:3

Enter second time in the format HH:MM:SS

4:58:59

NEW TIME

7:1:2

Program : 33

Aim :- Create a class publisher Derive class book from publisher with attribut from book attributes price and no of pages. write a program that display information about a Python book use base class constructor invocation and method overriding.

Procedure :-

```
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", "Taming Python By Programming",
            Java Jose', 200, 219)
s1.display()
```

Result: the program has been exuted. the output was verified.

Output

Title : Taming Python by Programming

Author : Jeeva Jose

Price : 288

Number of Pages : 219

Program : 34

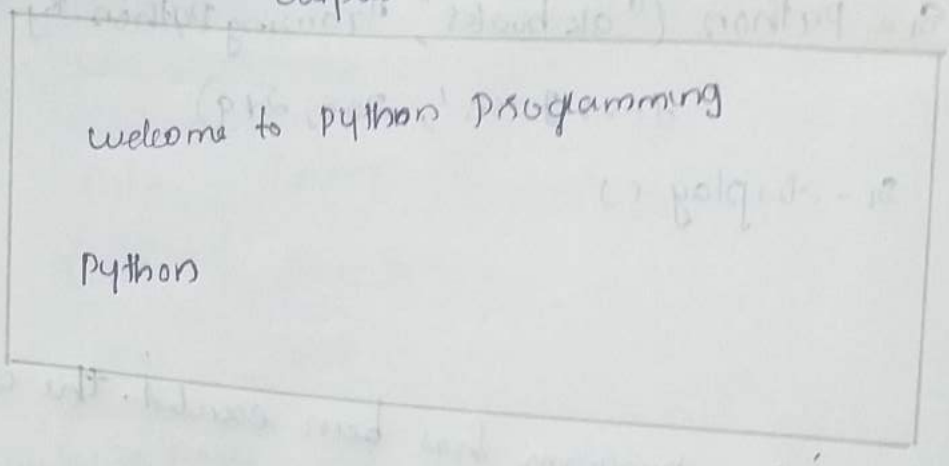
Aim : Write a Python program to read a file line by line and store it into a list

Procedure :

```
Str1 = "welcome to Python programming" "\n" "Python" "\n"
fw = open ("AFile.txt", "w")
fw.write (Str1)
fw.close ()
fr = open ("AFile.txt", "r")
str2 = fr.readlines ()
For i in str2:
print (i)
```

Result : The program was executed and has been successfully verified.

Output



Program : 35

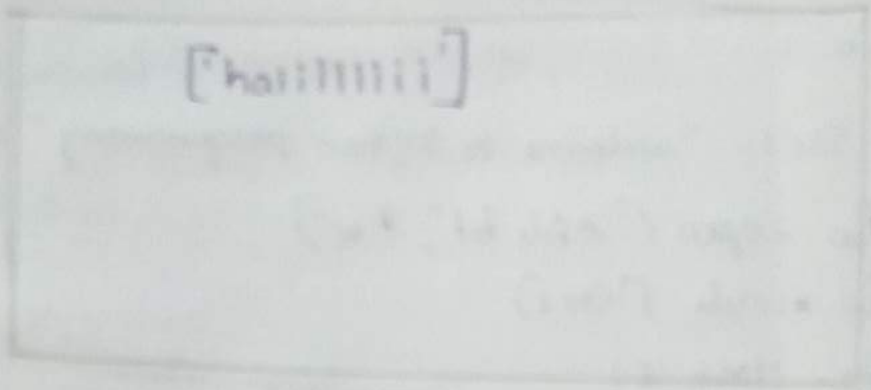
Aim : Python program to copy odd lines of one file to another.

Procedure :

```
fo = open ("file1.txt", 'r')
str1 = fo.readlines()
fo.close()
fo.open ("file2.txt", 'w')
x = 0;
for i in str1:
x = x+1
if x%2 != 0:
fo.write(i)
fo = open ("file2.txt", 'r')
str2 = fo.readlines()
Print (str2)
```

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

Output



Program: 36

Aim: Write a python program to read each row from a given csv file and print as list of strings.

Procedure:

```
Import csv
with open ('movie1.csv', 'w', newline = '')
as file:
writer = csv.writer (file)
writer = writer.row (['SN', 'movie', 'Ratings'])
writer.writerow ([1, 'Lord of Rings', 5])
writer.writerow ([2, 'Harry Potter', 6])
with open ('movie1.csv') as csv file:
    data = csv.reader (csv file)
    For row in data:
    print (','.join (row))
```

Result: The above program was executed and the result was obtained.

Output

Sn, movie, Rating

1) Lord of the rings, 5

2. Harry Potter

Program: 37

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

procedure:

```
import csv
with open ("fruit1.csv",'w', newline: '') as file :
write = csv.writer(file)
write.writerow (["slno", "fruits", "rate"])
write.writerow (["1", "apple", "60"])
write.writerow (["2", "orange", "55"])
write.writerow (["3", "grapes", "70"])
write.writerow (["4", "banana", "85"])
with open ("fruit1.csv",'r') as file:
data = csv.reader (file)
print ("content in column fruit:")
for i in data:
print (i[1])
```

Result: the above program was successfully executed and the result verified.

Output

```
Contents in the column 'fruits':

Fruits
apple
orange
grapes
banana
```

Program: 38

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

Procedure:

```
import csv
f = open("fruits.csv", "w")
writer = csv.DictWriter(f, fieldnames=["fruit", "cow"])
writer.writeheader()
writer.writerow({"fruit": "Apple", "count": "1"})
writer.writerow({"fruit": "Banana", "count": "2"})
f.close()
c=0
f.open("fruits.csv")
reader = csv.DictReader(f)
for row in header:
    if c=0
    print{" ".join(row)}')
    print(f'{row["fruit"]}, {row["count"]}')
    f.close()
```

Result:- The program was successfully executed and the result was verified.

Output

Fruit	count
Apple,	1
Fruit	count
Banana,	2

Program: 39

Aim: Create a package graphics with modules rectangle, circle and such package 3D-graphics with modules (cuboid and sphere) include method to find area and perimeter of different respective figures in each module. write program that finds area and perimeter of figures by different importing statements.

procedure:

```
from graphics.rectangle import *
from graphics.write import *
from graphics.3dgraphics.sphere import *
from graphics.3dgraphics.cuboid import *
print("Rectangle")
l = int(input("enter length:"))
b = int(input("enter breadth:"))
print("Area of Rectangle:",area(l,b))
print("perimeter of Rectangle:",perimeter(l,b))
print("circle")
r = int(input("enter a radius:"))
print("area of circle:",area(r))
print("circumference of circle:",perimeter(r))
print("sphere")
r = int(input("enter a radius:"))
print("area of sphere:",3*area(r))
print("int perimeter of sphere:",3*perimeter(r))
print("cuboid")
l = int(input("enter a
```

```
w = int (input ("enter a width"))
h = int (input ("enter a height:"))
print ("area of cuboid :", cuarea (l,w,h))
print ("permeter of cuboid:", cupermeter (l,w,h))

# circle
from math import pi
def carea (r):
  return (pi * r * r)
def cperimeter (r):
  return (2 * pi * r)

# rectangle
def rarea (l, b):
  return (l * b)
def rperimeter (l, b):
  return (2 * l + b))

# sphere
from math import pi
  def sarea (r)
  return (4 * pi * r * r)
def sperimeter (r):
  return ((4/3) * pi * r * r * r):

# cuboid
  def carea (l, w, h)
  return ((2 * l * w) + (2 * l * w) + (2 * w * h))
  def cuperimeter (l, w, h)
  return (4 * (l * w * h))
```

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

Output

Rectangle
Enter Length : 5
Enter Breadth : 10
Area of rectangle : 50
perimeter of rectangle : 30

Circle
Enter Radius of Circle : 4
Area of Circle : 50.26548245
perimeter of Circle : 25.13274172 8.

Sphere
Enter the Radius of sphere : 3
Area of Sphere : 113.097335 29 23255
perimeter of sphere : 113.097335 29 2325

Cuboid :
Enter Length : 7
Enter width : 12
Enter height : 8
Area of cuboid : 472
perimeter of cuboid : 108.