**CYCLE-1**

**PROGRAM NUMBER-1**

**Aim-**

Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev…

**PROGRAM NUMBER-2**

**Aim** –

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

**Algorithm** –

1)Ask for a year

2) take starting year as 2021 and final year as the entered year

3)check each year divisible by 100

4)check each year divisible by 400

5)check each year divisible by 4

6) print all the years satisfies

**Program**-

n=int (input("enter the last year :"))

for i in range (2021,n+1):

   if i%100==0:

      if i%400==0:

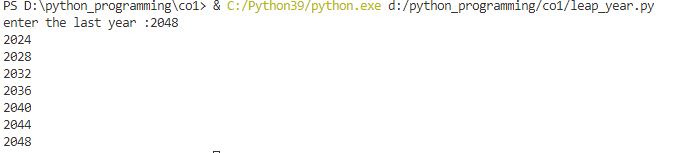
          print(i)

   else:

      if i%4==0:

         print(i)

**Output**-



**PROGRAM** **NUMBER**-3

**Aim**-

List comprehensions:

1. *Generate positive list of numbers from a given list of integers*

**Algorithm**-

a) Enter a list of numbers

b) check the positive number from the list

c)print all the positive numbers as list

**Program**-

n=[10,-89,-56,78,98,-8,-34]

l=[x for x in n if x>0 ]

print(l)

**Output**-



*b)Square of N numbers*

**Algorithm**-

1. Ask the user to enter a number
2. Print squares of numbers in a list

**Program**-

n=int(input("enter a value for n :"))

s=[x\*\*2 for x in range(n+1)]

print(s)

**Output**-



*c)Form a list of vowels selected from a given word*

**Algorithm**-

a) Enter a word

b) Check for the vowels from the word

c) Print the vowels from the word

**Program**-

l=[x for x in 'elephant ' if x in['a','e','i','o','u']]

print (l)

**Output**-



*d) List ordinal value of each element of a word*

**Algorithm**-

a)Enter a word

b)Print the ordinal value of each letter in the given word

**Program**-

w=input("enter a word: ")

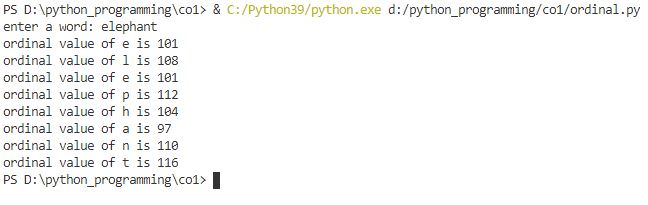
l=list(w)

for i in l:

    os=ord(i)

    print("ordinal value of",i,"is",os)

**Output**-



**PROGRAM** **NUMBER**-4

**Aim**-

Count the occurrences of each word in a line of text

**Algorithm**-

a)Enter a sentence

b)Split into words

c)Count each word to check the occurrence

d)Print the count

**Program**-

def sentence(str):

counts = dict()

words = str.split()

for i in words:

if i in counts:

counts[i] += 1

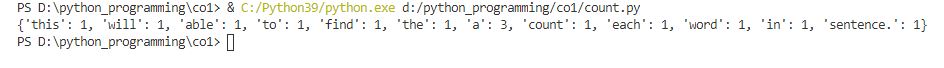
else:

counts[i] = 1

return counts

print( sentence('this will able to a find the a count character in a sentence.'))

**Output**-



**PROGRAM** **NUMBER**-5

**Aim**-

Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead

**Algorithm**-

a)Enter a list of numbers

b)Check for numbers greater than 100

c)Swap the number with “over” instead

d)Reprint the elements

**Program**-

li=[200,1,2,390,23,100]

for i in li:

    if i >=100:

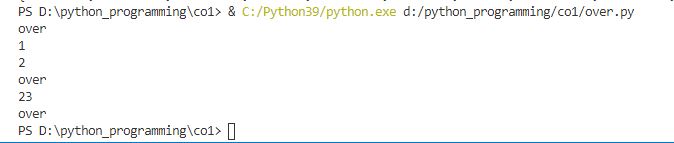
        i="over"

    else:

        i=i

    print(i)

**Output**-



**PROGRAM** **NUSMBER**-6

**Aim**-

Store a list of first names. Count the occurrences of ‘a’ within the list

**Algorithm**-

a)Enter a list of names

b)Take each name and count “a”

c)Print the count

Program-

l=["arya","lakshmi","albin","ann"]

t=0

for i in l:

    t+=i.count("a")

print("occurance of 'a': ",t)

**Output**-



**PROGRAM** **NUMBER**-7

**Aim**-

Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both

**Algorithm**-

a)Enter two lists of integers

b)Calculate the length of both lists

c)Check whether they have same length

d)If so print same length otherwise print not same length

e)Add the elements in each lists

f)Check whether the both lists have same added value

g)If so print same added value otherwise print not same added value

h)Traverse through each element in each elements

i)Check for common element

j)Print the common element

**Program**-

l1=[1,2,3,4]

l2=[1,3,4,5,6,7]

print('the 1st list is : ',l1)

print('the 2nd list is : ',l2)

leh1=len(l1)

leh2=len(l2)

if leh1==leh2:

    print("both lists are having same length")

else:

    print("both lists are not having same length")

k=0

for i in range(0, len(l1)):

    k = k + l1[i]

print ("the sum of l1: ",k)

l=0

for j in range(0, len(l2)) :

    l = l + l2[j]

print("the sum of l2: ",l)

if l==k:

    print("the sum of both lists are same")

else :

    print("the sum of both lists are not same")

def common(l1,l2):

    for i in l1:

        for j in l2:

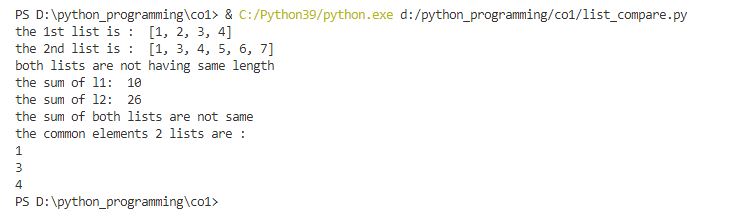
            if i==j:

                print(i)

print("the common elements 2 lists are : ")

common(l1,l2)

**Output**-



**PROGRAM** **NUMBER**-8

**Aim**-

Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character

**Algorithm**-

a)Get a word

b)Check for the first character of the word

c)Now check whether the first character is repeating or not

d) If so replace the repeated first character with “$”

e)Reprint the word with replaced symbol

**Program**-

w=input("enter your word : ")

c=w[0]

w=w.replace(c,"$")

w=c+w[1:]

print("your result is",w)

**Output**-



**PROGRAM NUMBER-9**

**Aim**-

Create a string from given string where first and last characters exchanged.

**Algorithm**-

a)Get a string of word

b)Calculate the length of the word

c)Split the first element and last element of the word

d)Swap the 2 character’s positions

e)Print the word

**Program**-

w=(input("enter your word: "))

length=len(w)

result=w[-1]+w[1:length-1]+w[0]

print(result)

**Output**-



**PROGRAM NUMBER-10**

**Aim**-

Accept the radius from user and find area of circle.

**Algorithm**-

a)Get a value for radius

b)Calculate the area of a circle (3.14\*r\*r)

c)Print the radius

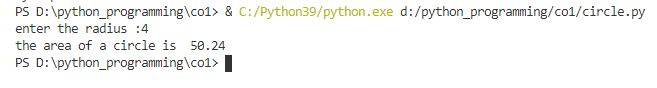
**Program-**

n=int(input("enter the radius :"))

circle=3.14\*n\*n

print("the area of a circle is ",circle**)**

**Output**

****

**PROGRAM NUMBER-11**

**Aim**-

Find biggest of 3 numbers entered.

**Algorithm**-

a)Enter 3 numbers

b)Compare 3 number using max function

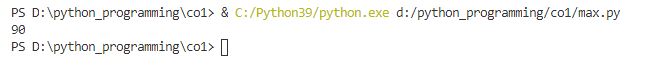
c)print the biggest number

Program

l=(7,5,6,90)

print(max(l))

**Output**-



**PROGRAM** **NUMBER**-12

**Aim**-

Accept a file name from user and print extension of that

**Algorithm**-

a)Enter a filename

b)Split the name and file type

c)Print the extension

**Program**-

n=input ("enter the filename : ")

f\_exe=n.split(".")

print("the extension of the file is : ",f\_exe[-1])

**Output**-



**PROGRAM NUMBER-13**

**Aim**-

Create a list of colors from comma-separated color names entered by user. Display first and last colors.

**Algorithm**-

a)Enter a list of colours

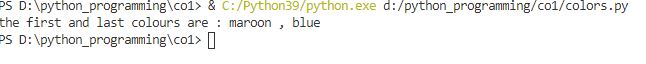
b)print the first and last elements in the list

**Program**-

n=["maroon","black","green","yellow","blue"]

print("the first and last colours are :",n[0],",",n[-1])

**Output-**



**PROGRAM NUMBER-14**

**Aim-**

Accept an integer n and compute n+nn+nnn.

**Algorithm-**

a)ask the user to enter a number

b)take the number as n

c)substitute in the equation (n+n\*n+n\*n\*n)

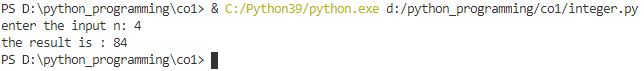
d)print the result

**Program-**

n=int(input("enter the input n: "))

print("the result is :",n+n\*n+n\*n\*n)

**Output-**



**PROGRAM NUMBER-15**

**Aim-**

Print out all colors from color-list1 not contained in color-list2.

**Algorithm-**

a)enter 2 lists of colours

b)find the colours not contained in second list using difference function

c)print those colours

**Program-**

color1=set(['orange','blue','black','violet','purple'])

color2=set(['blue','violet'])

print(color1.difference(color2))

**output-**



**PROGRAM NUMBER-16**

**Aim-**

Create a single string separated with space from two strings by swapping the character at position 1.

**Algorithm-**

a)Enter a string of 2 words

bCcreate a function named swapping

c)It swap the first letters of both words among themselves

d)Print the string after swapping

**Program-**

def swapping(a,b):

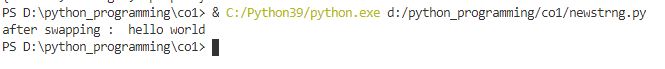
    a1=b[:1]+a[1:]

    b1=a[:1]+b[1:]

    print("after swapping : ",a1+" "+b1)

swapping('wello' , 'horld')

**Output-**



**PROGRAM NUMBER-17**

**Aim-**

Sort dictionary in ascending and descending order.

**Algorithm-**

a)enter a dictionary

b)sort the dictionary in ascending and descending order using operator moudule

c)use itemgetter function to sort

d)print the dictionaries in both orders

**Program**-

import operator

k ={4:2,3:1,5:6,7:8}

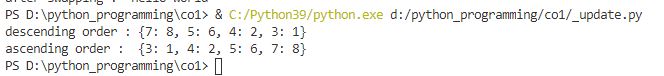
sorted\_kdes =dict(sorted(k.items(),key =operator.itemgetter(1),reverse=True))

print('descending order :',sorted\_kdes)

sorted\_kasc =dict(sorted(k.items(),key =operator.itemgetter(1)))

print("ascending order : ",sorted\_kasc)

**Output**-



**PROGRAM NUMBER-18**

**Aim-**

Merge two dictionaries

**Algorithm-**

a)Enter two dictionaries

b)Use update function to merge

c)Print the merged dictionaries

**Program-**

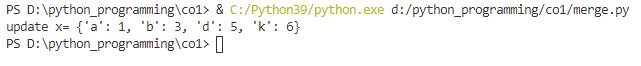
x={'a':1,'b':3}

y={'d':5,'k':6}

x.update(y)

print('update x=',x)

**Output-**



**PROGRAM NUMBER-19**

**Aim-**

Find gcd of 2 numbers.

**Algorithm-**

a)ask the user for 2 numbers

b)import math module

c)use gcd function

d)find the gcd of 2 numbers

e)print gcd

**Program-**

import math

a=int(input("enter your number : "))

b=int(input("enter your number : "))

print("gcd of the 2 digits is : ",math.gcd(a,b))

**Output-**



**PROGRAM NUMBER-20**

**Aim-**

From a list of integers, create a list removing even numbers.

**Algorithm-**

a)enter a list of numbers

b)use lambda function

c)use filter (built in function) from lambda function

d)put the condition for odd

e)filter out even numbers

f)print the list

**Program-**

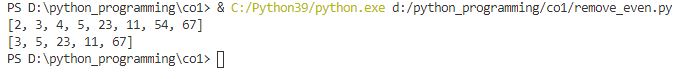
l=[2,3,4,5,23,11,54,67]

print (l)

even=list(filter(lambda x: (x%2 !=0),l))

print(even)

**Output-**



**CYCLE-2**

**PROGRAM NUMBER-1**

**Aim-**

Program to find the factorial of a number

**Algorithm**-

a)Enter the number

b)Put the base value

c)Put factorial(0)=1

d)Find the factorial for n (1\*2\*3\*…\*n)

e)Print the factorial result

**Program**-

n=int (input("enter the element :"))

factorial=1

if(n<0):

    print("sorry try non negative number")

elif(n==0):

    print("factorial of  0 is 1")

else:

    for i in range (1,n+1):

        factorial=factorial\*i

print("the factorial of",n,"is :",factorial)

**Output**-



**PROGRAM NUMBER-2**

**Aim-**

Generate Fibonacci series of N terms

**Algorithm-**

a)Enter a number(number of fibonacci elements)

b)Check if it is positive or not

c)If not ask for positive

d)Create a function that gives the Fibonacci numbers

e)Print the fibonacci numbers

**Program-**

def fib(num):

    if num <= 1:

        return num

    else:

        return(fib(num-1)+fib(num-2))

n=int(input("enter the limit: "))

if n<=0:

    print("enter a positive number")

else:

    print("the fibonnacii series is : ")

    for i in range (n):

        print(fib(i))

**Output**-



**PROGRAM NUMBER-3**

**Aim-**

Find the sum of all items in a list

**Algorithm-**

a)Enter a list

b)Import functiontools

c)Use reduce function by using lambda

d)Add all the elements in the list

e)Print the sum

**Program-**

import functools

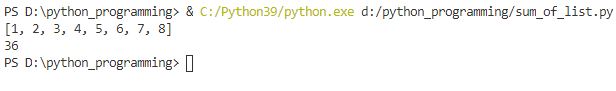
l=[1,2,3,4,5,6,7,8]

print(l)

n=functools.reduce(lambda x,y:x+y,l)#reduce function

print (n)

**Output-**



**PROGRAM NUMBER-4**

**Aim**-

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

**Algorithm-**

a)Set the range

b)Square the numbers such that the product is 4 digit

c)Check the number is even or not

d)If even print the square

**Program**-

for i in range(1000,10000):

 for j in range(32,100):

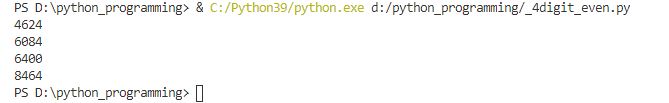
      if i==j\*j:

       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)

**Output**-



**PROGRAM NUMBER-5**

**Aim-**

Display the given pyramid with step number accepted from user.

**Algorithm-**

a)Ask for step number

b)Create a function

c)Prints the numbers in a pyramid with steps received from user

**Program-**

def countnum(n):

    num=1

    for i in range(0,n):

        for j in range (0,i+1):

            print(num,end=' ')

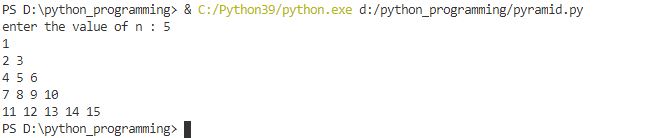
            num=num+1

        print("\r")

n=int(input("enter the value of n : "))

countnum(n)

**Output-**

****

**PROGRAM NUMBER-6**

**Aim**-

Count the number of characters (character frequency) in a string

**Algorithm-**

a)Enter a sentence

b)Create a function

c)Counts each character in a string

d)Print count in a dictionary

**Program-**

def sentence(str):

    count={}

    for i in str:

        if i in count:

            count[i] += 1

        else:

            count[i] = 1

    return count

print( sentence("grphsss"))

**Output-**



**PROGRAM NUMBER-7**

**Aim-1**

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

**Algorithm-**

a)Ask for a word

b)Create a function

c)Check the word

d)If it end with “ing” add “ly” to the word

e)If it doesn’t end with “ing” then add “ing”

f)Print the word after adding

**Program-**

def add(sent):

    length=len(sent)

    if length>2:

        if sent[-3:]=='ing':

            sent += 'ly'

        else:

             sent+=  'ing'

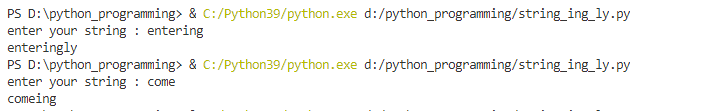
    return sent

sent=input("enter your string : ")

add(sent)

print(add(sent))

**Output-**



**PROGRAM NUMBER-8**

**Aim**-

Accept a list of words and return length of longest word.

**Algorithm-**

a)Enter words

b)Create a function

c)Function counts length

d)Compare the length

e)Print the longest

**Program-**

def longest\_word(w):

    word=len(w[0])

    length=w[0]

    for i in w:

        if(len(i)>word):

            word=len(i)

            length=i

    print("the longest word is : ",length)

w=['kamaggumonna','tommy','worsinija']

longest\_word(w)

**Output-**



**PROGRAM NUMBER-9**

**Aim-**

Construct following pattern using nested loop

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Algorithm-**

a)Enter a number

b)Create a function

c)Use nested loop to get a increasing pyramid

d)Use nested loop to get a decreasing pyramid

e)Print the star pyramid

**Program-**

def star(n):

    for i in range (0,n):

        for j in range(0,i+1):

            print("\*",end=" ")

        print("\r")

    for i in range(n,0,-1):

        for j in range(i+1,0,-1):

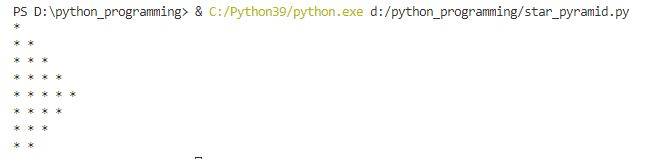
           print("\*",end=" ")

        print("\r")

n=4

star(n)

**Output**-



**PROGRAM NUMBER-10**

**Aim-**

Generate all factors of a number.

**Algorithm-**

a)Take a number from user

b)Create a function

c)it checks the numbers divide the given number

d)print all the factors

**Program-**

def fac(n):

    for i in range (1,n+1):

        if n %i==0:

            print(i)

n=int(input("enter the number whose factors to be found : "))

fac(n)

**Output**-



**PROGRAM NUMBER-11**

**Aim-**

Write lambda functions to find area of square, rectangle and triangle.

**Algorithm-**

a)Ask user for side of a square

b)Create lambda function to find the area of a square

c)Print the area

d)Ask the user for length and breadth of a rectangle

e)Create lambda function to find the area of the rectangle

f)Print the area

g)Ask the user for base and height of a triangle

h)Create lambda function to find the area of the triangle

i)Print the area

**Program-**

square=lambda x: x\*x

a=int(input("enter the side : "))

print("area of  square is : ",square(a))

rectangle=lambda a,b: a\*b

a=int(input("enter the length : "))

b=int(input("enter the breadth : "))

print("area of rectangle is : ",rectangle(a,b))

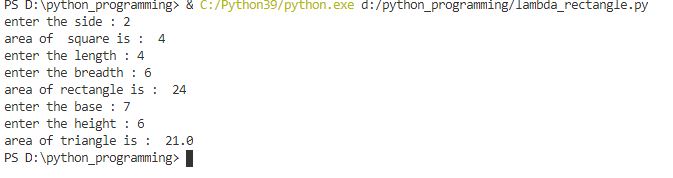
triangle =lambda b,h:(1/2)\*b\*h

b=int(input("enter the base : "))

h=int(input("enter the height : "))

print("area of triangle is : ",triangle(b,h))

**Output-**



**CYCLE-3**

**PROGRAM NUMBER-1**

Aim-

Work with built-in packages

PROGRAM NUMBER-2

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)

Algorithm-

**CYCLE-4**

**PROGRAM NUMBER-1**

**Aim-**

Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

**Algorithm-**

a)Enter the length and breadth

b)Create a class named “rectangle”

c)Create functions to find area inside the class

d)Create function to find perimeter inside the class

e)To find the area call the area function from class “rectangle”

f)To find the perimeter call the perimeter function from class “rectangle”

g) Compare the areas

i)Print the results

**Program-**

class rectangle:

def \_\_init\_\_(self,length,breadth):

self.length=length

self.breadth=breadth

def area(self):

return self.length\*self.breadth

def perimeter(self):

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

a=int(input("length of rectangle1 : "))

b=int(input("breadth of rectangle1 : "))

c=int(input("length of rectangle2 : "))

d=int(input("breadth of rectangle2 : "))

rec1=rectangle(a,b)

rec2=rectangle(c,d)

print("area of rectangle1 : ",rec1.area())

print("perimeter of rectangle1 : ",rec1.perimeter())

print("area of rectangle2 : ",rec2.area())

print("perimeter of rectangle2 : ",rec2.perimeter())

if rec1.area()==rec2.area():

print("the 2 given rectangle have same area")

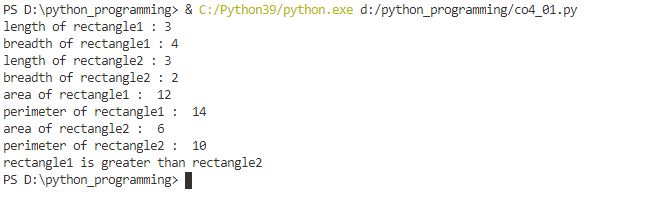
elif rec1.area() > rec2.area():

print("rectangle1 is greater than rectangle2")

else:

print("rectangle2 is greater than rectangle1")

**Output-**



**PROGRAM NUMBER-2**

**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.

**Algorithm-**

a)Create a class named “bank”

b)Create a deposit function

c)Create a withdraw function

d)As the first step let the user create an account

e)Ask the user to deposit some amount use deposit function

f)User can withdraw an amount of money

g)Checks whether the amount is available or not

h)If yes then transaction will be successful

i)Print the withdraw amount

j)If no then print insufficient amount

**Program-**

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("you have withdrew : ",amount)

        else:

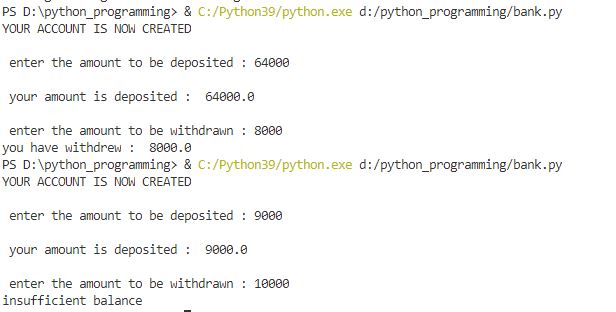
            print("insufficient balance ")

a=bank()

a.deposit()

a.withdraw()

Output-



**PROGRAM NUMBER-3**

**Aim-**

Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.

**Algorithm-**

a)Create a class “A”

b)Create a area function to find the area

c) Include \_\_gt\_\_ function to compare and return the greatest rectangle

d)Enter the first rectangle with length and breadth

e)Call the area function to find the area

f)Enter the 2nd rectangle with length and breadth

g)Call the area function again to find the area

h)Now compare the 2 rectangles

i)print the result

**Program-**

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

rect1=A(3,4)

rect1.area()

rect2=A(6,5)

rect2.area()

if(rect1>rect2):

    print("rect1 is greater than rect2")

else:

    print("rect2 is greater than rect1")

**Output-**



**PROGRAM NUMBER-4**

**Aim**-

Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.

**Algorithm**-

a)Create a class “Time”

b)Create a function of “convert” to convert

60 seconds or greater to 1minute

60 minutes or greater to 1 hour

c)Use “\_\_add\_\_”function inorder to add the time

d)enter 2 times

e)Call “\_\_add\_\_” function

f)Call “convert” function

g)Print the added and converted result

**Program-**

class Time:

    def \_\_init\_\_(self,h,m,s):

        self.\_\_hour=h

        self.\_\_minute=m

        self.\_\_seconds=s

    def convert(self):

        if self.\_\_seconds>=60:

            self.\_\_seconds-=60

            self.\_\_minute +=1

        if self.\_\_minute>=60:

            self.\_\_minute-=60

            self.\_\_hour +=1

        return(self.\_\_hour,self.\_\_minute,self.\_\_seconds)

    def \_\_add\_\_(self,other):

        print("Time1:",(self.\_\_hour,self.\_\_minute,self.\_\_seconds))

        print("Time2:",(other.\_\_hour,other.\_\_minute,other.\_\_seconds))

        self.\_\_hour=self.\_\_hour + other.\_\_hour

        self.\_\_minute=self.\_\_minute + other.\_\_minute

        self.\_\_seconds=self.\_\_seconds + other.\_\_seconds

        return(self)

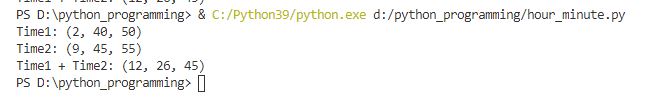
t1=Time(2,40,50)

t2=Time(9,45,55)

t3=(t1+t2)

print("Time1 + Time2:",t3. convert())

**Output-**



**PROGRAM NUMBER-5**

**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 overriding.

**Algorithm-**

a)Create a class “publisher”

b)Derive a class named “book” from “publisher”

c) Create functions with attributes of title and author

d)Derive a class named “python” from “book”

e)Create functions with attributes price and number of pages

f)Ask for a publisher,book ,author, page numbers, price

g)Transfer these information to the “python”class

h)Using display function the information is printed

**Program-**

class publisher:

    def \_\_init\_\_(self,pname):

        self.pname=pname

    def display(self):

        print("name: ",self.pname)

class book(publisher):

    def \_\_init\_\_(self,pname,bname,author):

        self.pname=pname

        self.bname=bname

        self.author=author

    def display(self):

        print("pname: ",self.pname)

        print("bname: ",self.bname)

        print("author: ",self.author)

class python (book):

    def \_\_init\_\_(self,pname,bname,author,page,price):

        self.pname=pname

        self.bname=bname

        self.author=author

        self.page=page

        self.price=price

    def display(self):

        print("pname: ",self.pname)

        print("bname: ",self.bname)

        print("author: ",self.author)

        print("page: ",self.page)

        print("price: ",self.price)

n=input("enter publisher:")

b=input("enter book name:")

t=input("enter author name: ")

p=int(input("enter page number : "))

pr=int(input("enter price: "))

print("------------------------------------------------------------------------")

obj=python(n,b,t,p,pr)

obj.display()

**Output-**



**CYCLE-5**

**PROGRAM NUMBER-1**

**Aim**

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

Algorithm-

a)Enter strings

b)Open a txt file which can be written

c)Copy the strings to the text file

d)Close the file

e)Again open the new file created in read only format

f)Print the content from the new text file

Program-

st1="Good Morning""\n""hello everyone""\n""allright""\n"

fw=open("Afile.txt","w")

fw.write(st1)

fw.close()

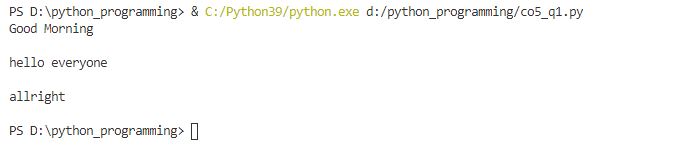
fr=open("Afile.txt","r")

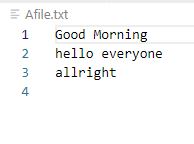
st2=fr.readlines()

for i in st2:

    print(i)

Output-





**PROGRAM NUMBER-2**

**Aim**-

Pyhon program to copy odd lines of one file to other

**Algorithm-**

a)Create a file with some content

b)Open this file with read mode

c)Read each lines

d)Close the file

e)Open a new file in write mode

f)Select the odd lines from the first text document

g)Copy that lines to the new document

**Program-**

f = open("sample.txt",'r')

str1 = f.readlines()

f.close()

f = open("s2.txt",'w')

x = 0;

for i in str1:

    x = x+1

    if x%2!=0:

        f.write(i)

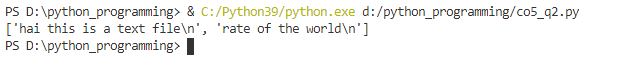
f.close()

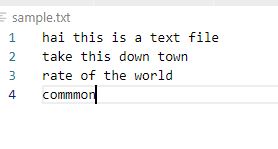
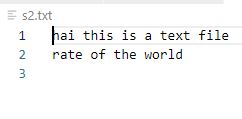
f=open("s2.txt",'r')

str2=f.readlines()

print(str2)

**Output-**





PROGRAM NUMBER-

Aim-

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

Algorithm-

a)import a csv

b)create a csv names “movie.csv”

c)