**COURSE OUTCOME 1**

**PROGRAM NO: 1**

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

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

* An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
* Build, execution, and debugging tools
* Some form of source control

Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly.

**General Editors and IDEs with Python Support**

* **Eclipse + PyDev**

**Category:** IDE  
**Website:** [www.eclipse.org](http://www.eclipse.org/)  
**Python tools:** PyDev, [www.pydev.org](https://www.pydev.org/)

If you’ve spent any amount of time in the open-source community, you’ve heard about Eclipse. Available for Linux, Windows, and OS X at, Eclipse is the de-facto open-source IDE for Java development. It has a rich marketplace of extensions and add-ons, which makes Eclipse useful for a wide range of development activities.

One such extension is PyDev, which enables Python debugging, code completion, and an interactive Python console. Installing PyDev into Eclipse is easy: from Eclipse, select Help, Eclipse Marketplace, then search for PyDev. Click Install and restart Eclipse if necessary.

### Sublime Text

**Category:**CodeEditor  
 **Website:** [http://www.sublimetext.com](http://www.sublimetext.com/)

Written by a Google engineer with a dream for a better text editor, Sublime Text is an extremely popular code editor. Supported on all platforms, Sublime Text has built-in support for Python code editing and a rich set of extensions (called packages) that extend the syntax and editing features.Installing additional [Python packages](https://realpython.com/python-modules-packages/) can be tricky: all Sublime Text packages are written in Python itself, and installing community packages often requires you to execute Python scripts directly in Sublime Text.

### Visual Studio

**Category:** IDE  
**Website:** <https://www.visualstudio.com/vs/>  
**Python tools:** [Python Tools for Visual Studio](http://pytools.codeplex.com/), aka PTVS

Built by Microsoft, Visual Studio is a full-featured IDE, in many ways comparable to Eclipse. Built for Windows and Mac OS only, VS comes in both free (Community) and paid (Professional and Enterprise) versions. Visual Studio enables development for a variety of platforms and comes with its own marketplace for extensions.Python Tools for Visual Studio (aka PTVS) enables Python coding in Visual Studio, as well as Intellisense for Python, debugging, and other tools.

## Python-Specific Editors and IDEs

### PyCharm

**Category:** IDE  
**Website:** <https://www.jetbrains.com/pycharm/>

One of the best (and only) full-featured, dedicated IDEs for Python is [PyCharm](https://realpython.com/pycharm-guide/). Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.

**PROGRAM NO: 2**

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

**ALGORITHM:**

Step1: Take current year and final year as inputs

Step2: if current year < final year

Step3: Check for leap year condition

Step4: Then print list of leap years between current and final year

**PROGRAM:**

current = int(input("Enter current year: "))

final = int(input("Enter final year: "))

if current < final:

print ("Here is a list of leap years between " + str(current) + " and " + str(final) + ":")

while current < final:

if current % 4 == 0:

print(current)

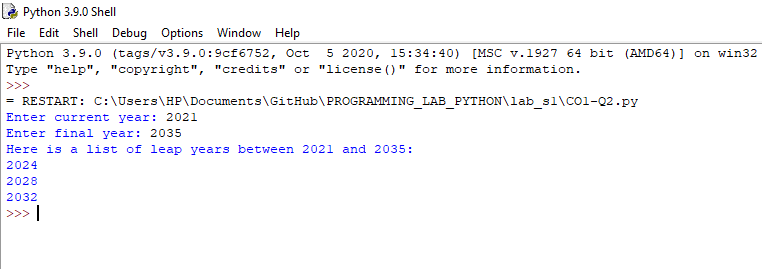
if current % 100 == 0 and current % 400 == 0:

print(current)

current += 1

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT:**

****

**PROGRAM NO: 3**

**AIM**: List comprehensions:

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

(b) Square of N numbers

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

(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

**ALGORITHM:**

1. Step1: Take in the number of elements to be in the list from the user.

Step2: Using a for in list comprehensions, get the elements one by one from the list and check if it is positive

Step3: If it is positive, print the numbers as a list

1. Step1:

**PROGRAM:**

a)

list1 = [1,-1, -21, 0, 45, 66,2,-3,4,-6, -93]

print(list1)

a=[num for num in list1 if num>=0]

print("Positive integers in the list are:",a)

b)

N=int(input("Enter limit N:"))

x = (x\*\*2 for x in range(N))

x = list(x)

print(x)

c)

V =[' a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U' ]

print("V=['a', 'e', 'i', 'o', 'u'', ‘A', 'E', 'I', 'O', 'U' ]")

w=str(input("Enter the word: "))

x = [x for x in w if any([v in x for v in V])]

x = list(x)

print ("Vowels in given word:",x)

d) a=str(input("Enter word:"))

a=list(a)

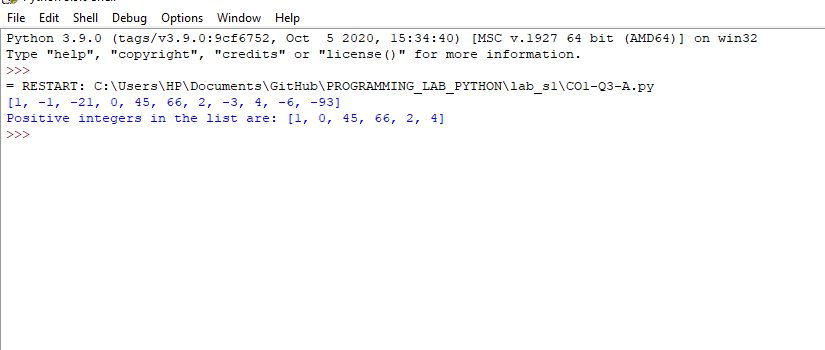
x=[ord(x) for x in a ]

x=list(x)

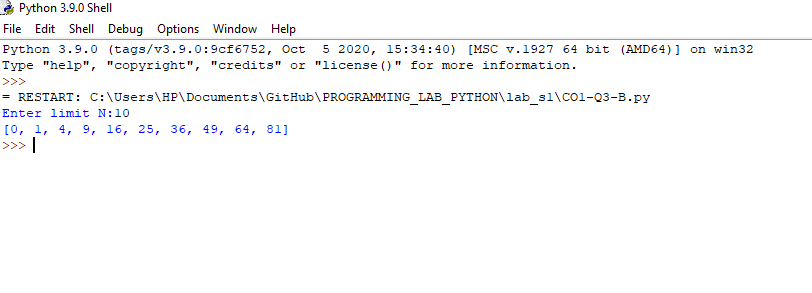
print(x)

**RESULT:** The above program is successfully executed and obtained the output

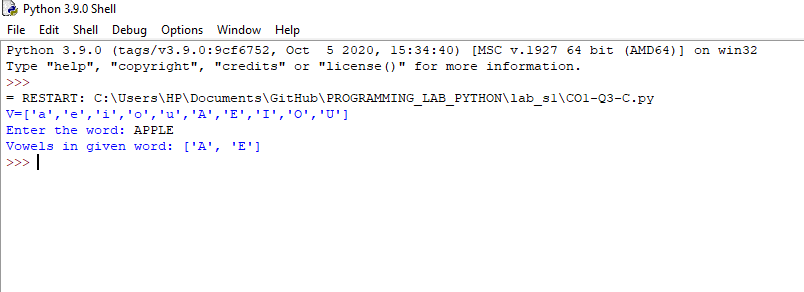
**OUTPUT:**

**a)**

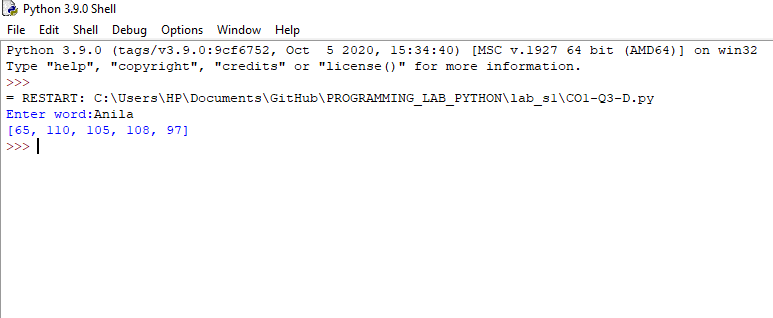
**b)**



**c)**



**d)**



**PROGRAM NO: 4**

**AIM**: Count the occurrences of each word in a line of text.

**ALGORITHM:**

**PROGRAM:**

a=str(input("Enter word:"))

print(a)

s=a.split(' ')

count = {}

print

for n in s:

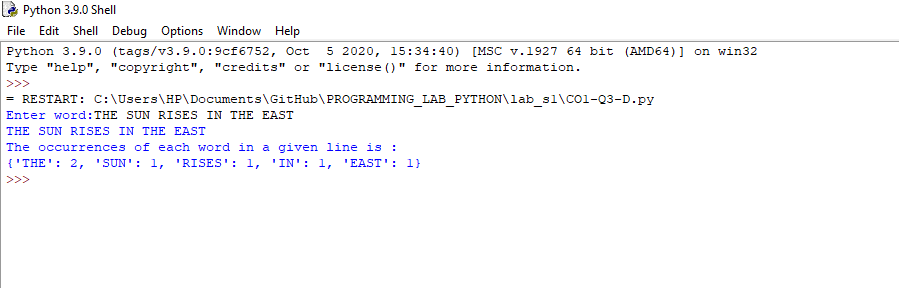
count[n]=count.get(n,0)+1

print("The occurrences of each word in a given line is :")

print(count)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 5**

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

**ALGORITHM**

**PROGRAM:**

lst = [ ]

lst = [int(item) for item in input("Enter the list items : ").split()]

print("INPUT IS",lst)

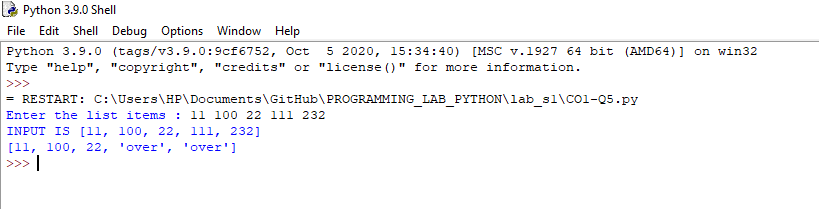
x= ["over" if x>100 else x for x in lst]

lst=list(x)

print(lst)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 6**

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

**ALGORITHM**

PROGRAM:

lst=['anu','ammu','ananya']

print(lst)

i=0

count=0

while i<len(lst):

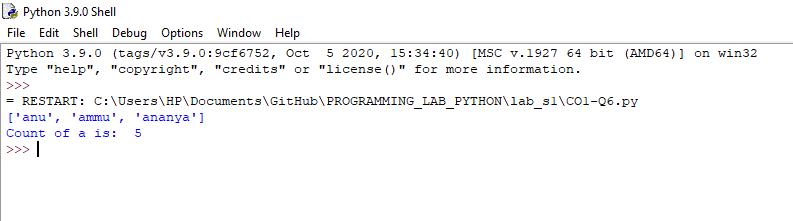
count=count+lst[i].count('a')

i=i+1

print("Count of a is: " ,count)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 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**

**PROGRAM:**

lst1=[4,5,3,2,1]

lst2=[8,4,3,2,1,5,9]

print("lst1=",lst1)

print("lst2=",lst2)

a=len(lst1)

b=len(lst2)

if a==b:

print("SAME LENGTH")

else:

print("NOT SAME LENGTH")

s1=sum(lst1)

s2=sum(lst2)

if s1==s2:

print("SUM IS SAME")

else:

print("SUM IS NOT SAME")

lst1=set(lst1)

lst2=set(lst2)

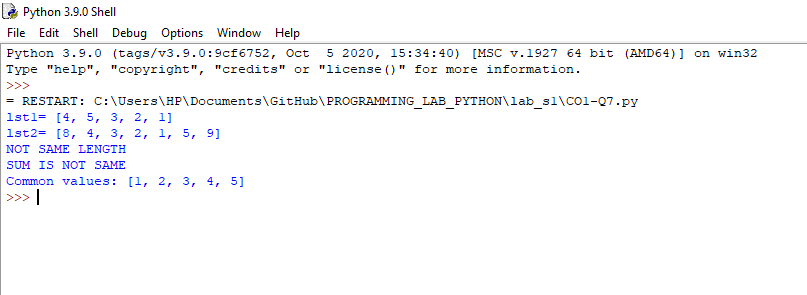
i = lst1.intersection(lst2)

i=list(i)

print("Common values:",i)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 8**

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

**ALGORITHM**

**PROGRAM:**

str1=input('Enter string ')

print('input sring is ',str1)

char = str1[0]

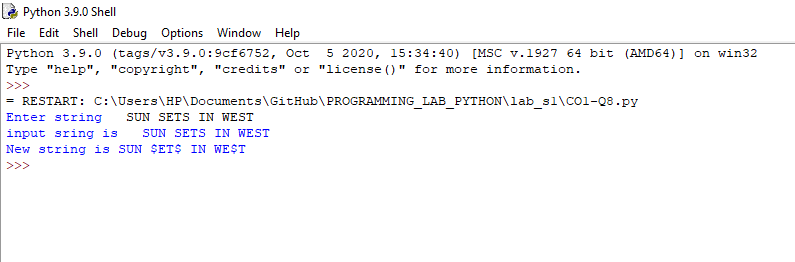
str1 = str1.replace(char, '$')

str1 = char + str1[1:]

print('New string is',str1)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 9**

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

**ALGORITHM**

**PROGRAM:**

s = str(input("Enter the String:"))

print("INPUT IS:",s)

slice\_mid=s[1:-1]

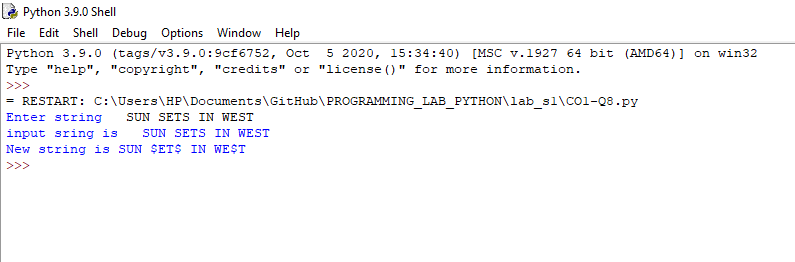
slice\_beg=s[0]

slice\_end=s[-1:]

print("Resultant string is:",slice\_end+slice\_mid+slice\_beg)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 10**

**AIM:** Accept the radius from user and find area of circle.

**ALGORITHM**

**PROGRAM:**

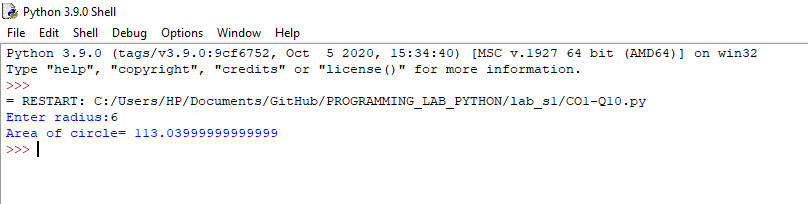
pi=3.14

R=int(input("Enter radius:"))

print("Area of circle=",pi\*R\*R)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 11**

**AIM:** Find biggest of 3 numbers entered.

**ALGORITHM**

**PROGRAM:**

n1 = float(input("Enter num 1:"))

n2 = float(input("Enter num 2:"))

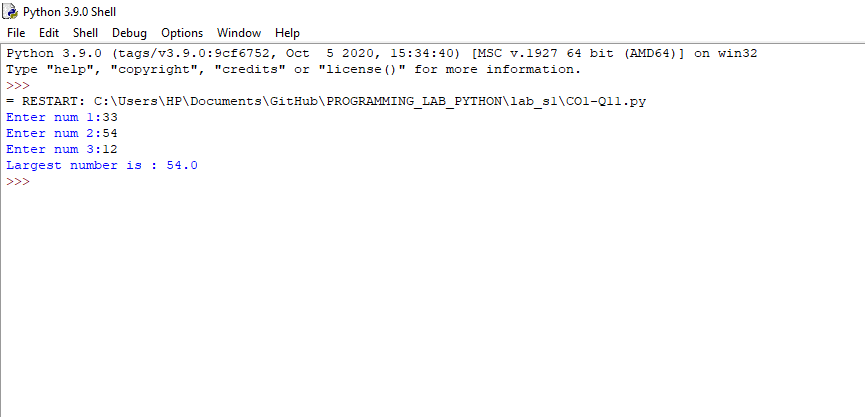
n3 = float(input("Enter num 3:"))

large=max(n1 , n2 , n3)

print("Largest number is :",large)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 12**

**AIM:** Accept a file name from user and print extension of that

**ALGORITHM**

**PROGRAM:**

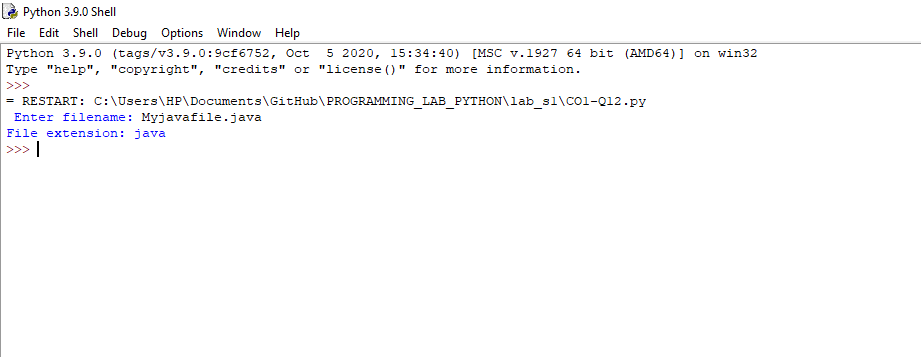
str=input(" Enter filename: ")

t=str.split('.')

print("File extension: " +t[-1])

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 13**

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

**ALGORITHM**

**PROGRAM:**

color\_lst=["red","blue","black","white","yellow","orange"]

print(color\_lst)

a = color\_lst[0]

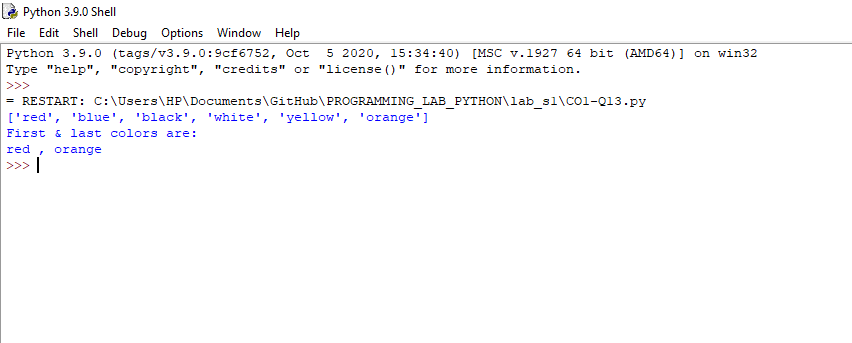
b= color\_lst[-1]

print("First & last colors are:")

print(a,b, sep = " , ")

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 14**

**AIM:** Accept an integer n and compute n+nn+nnn.

**ALGORITHM**

**PROGRAM:**

N = int(input("Enter the integer N :"))

tmp = N

tmp1 = tmp\*tmp

tmp2 = tmp\*tmp\*tmp

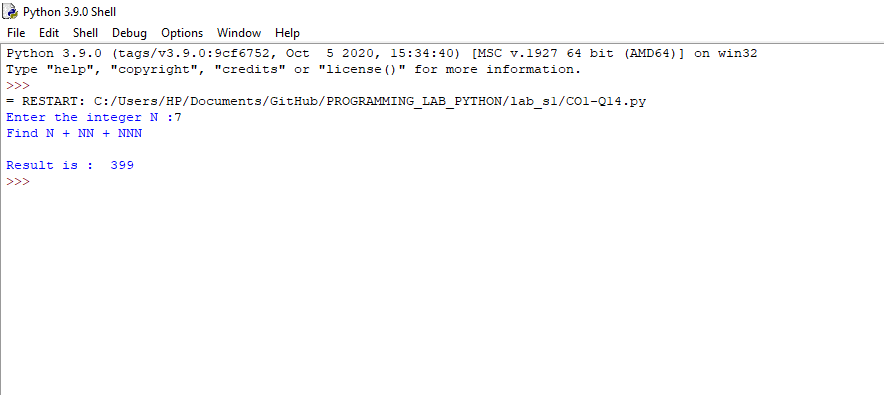
print("Find N + NN + NNN\n")

comp = tmp + tmp1 + tmp2

print("Result is : ",comp)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 15**

**AIM:** Print out all colors from color-list1 not contained in color-list2

**ALGORITHM**

**PROGRAM:**

colorlist1=set(['orange','green','blue','violet','pink','white'])

print(colorlist1)

colorlist2=set(['white','blue','violet'])

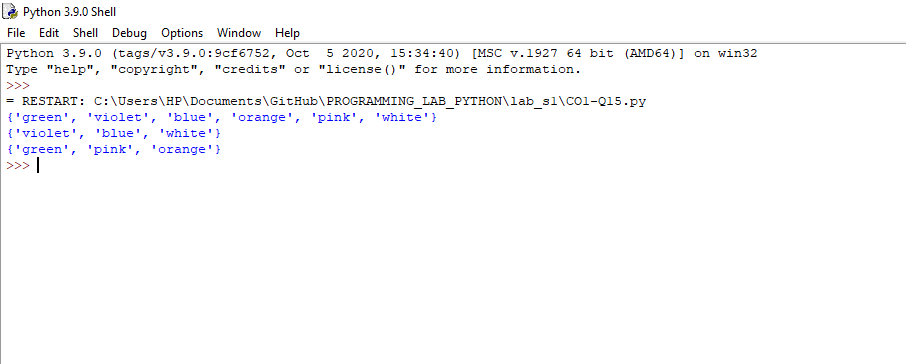
print(colorlist2)

a=(colorlist1.difference(colorlist2))

print(a)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 16**

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

**ALGORITHM**

**PROGRAM:**

a= "PYTHON"

b="JAVA"

print ("a=",a)

print("b=",b)

print(a +" " + b )

a1 = b[:1] + a[1:]

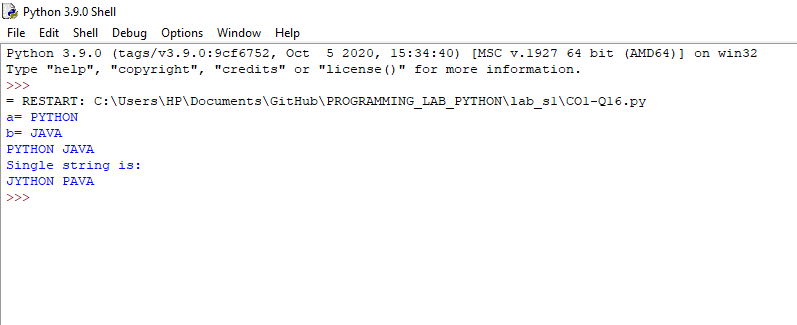
b2= a[:1] + b[1:]

print("Single string is:")

print(a1 + " "+ b2)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 17**

**AIM:** Sort dictionary in ascending and descending order.

**ALGORITHM**

**PROGRAM:**

a1 ={'Swathi':67,'Anu':98,'Riya':66,'Vismaya':88,'Neema':75,'Reshma':89}

print("Inputed dict is :", a1)

a1\_sorted\_keys = sorted(a1, key=a1.get, reverse=True)

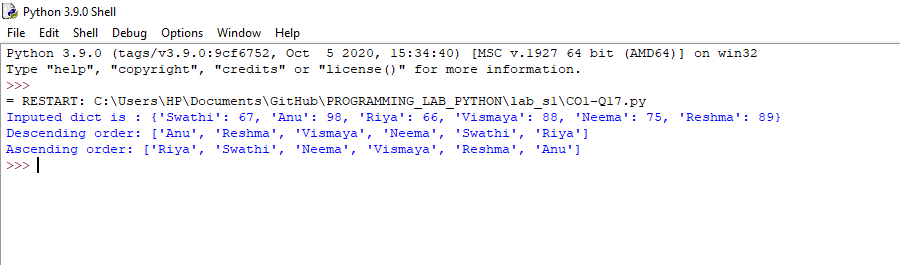
a1\_sorted\_keys\_2 = sorted(a1, key=a1.get)

print("Descending order:",a1\_sorted\_keys)

print("Ascending order:",a1\_sorted\_keys\_2)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 18**

**AIM:** Merge two dictionaries

**ALGORITHM**

**PROGRAM:**

def Merge(dict1, dict2):

return (dict2.update(dict1))

dict1 = {'a': 100, 'b': 48, 'e': 55}

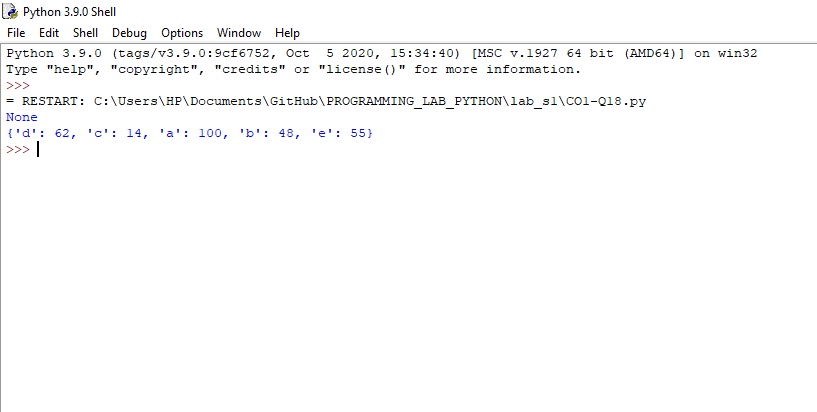
dict2 = {'d': 62, 'c': 14}

print(Merge(dict1, dict2))

print(dict2)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 19**

**AIM:** Find gcd of 2 numbers.

**ALGORITHM**

**PROGRAM:**

n1=int(input("ENTER FIRST NUM:"))

n2=int(input("ENTER SECOND NUM:"))

gcd=1

if n1%n2==0:

print(n2)

for k in range(int(n2 / 2), 0, -1):

if n1 % k == 0 and n2 % k == 0:

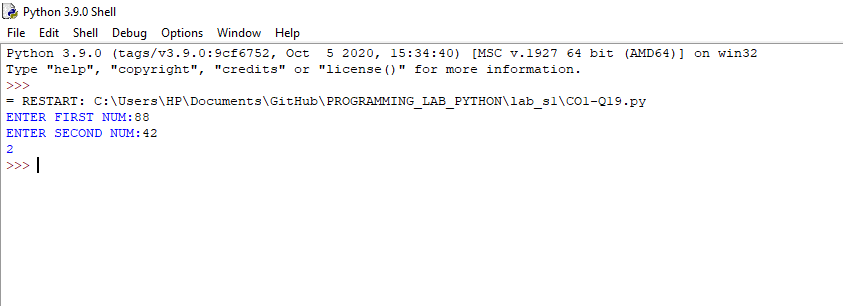
gcd = k

break

print(k)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 20**

**AIM:** From a list of integers, create a list removing even numbers.

**ALGORITHM**

**PROGRAM:**

li=[33,88,9,12,45,78,11,77]

print("ORIGINAL LIST:",li)

for i in li:

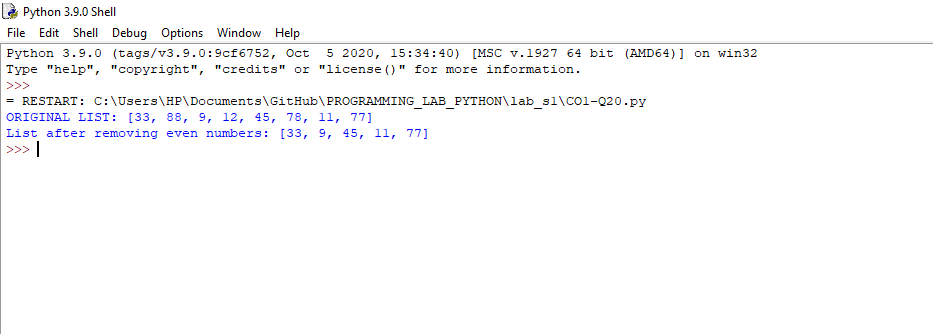
if (i % 2 ==0):

li.remove(i)

print("List after removing even numbers:",li)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**COURSE OUTCOME 2**

**PROGRAM NO: 1**

**AIM:** Program to find the factorial of a number

**ALGORITHM**

**PROGRAM:**

n=int(input("enter number:"))

fact=1

if n<0:

print("cannot find factorial")

elif n==0:

print("Factorial is 0")

else:

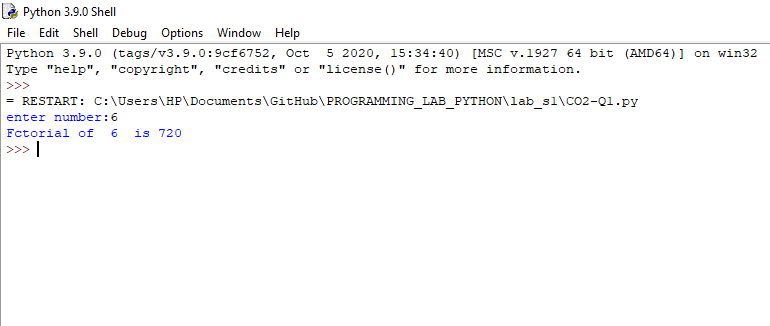
for i in range(1,n+1):

fact=fact\*i

print("Fctorial of ",n," is",fact)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 2**

**AIM:** Generate Fibonacci series of N terms

**ALGORITHM**

**PROGRAM:**

n=int(input("ENTER THE LIMIT:"))

f=0

s=1

if n<=0:

print("The requested series is",f)

else:

print(f,s,end=" ")

for x in range(2,n):

next=f+s

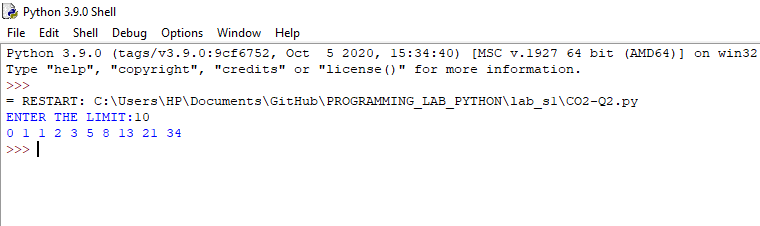
print(next,end=" ")

f=s

s=next

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 3**

**AIM:** Find the sum of all items in a list

**ALGORITHM**

PROGRAM:

a=[4,5,8,2,1,9]

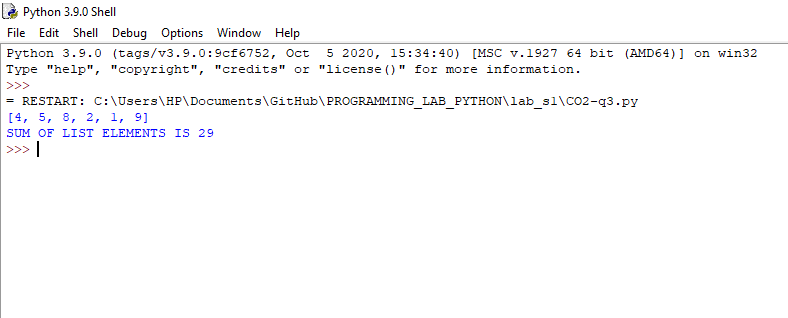
print(a)

b=sum(a)

print("SUM OF LIST ELEMENTS IS",b)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 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**

**PROGRAM**

F=int(input("Enter range from:"))

L=int(input("Enter range to:"))

a=[]

for x in range(F,L+1):

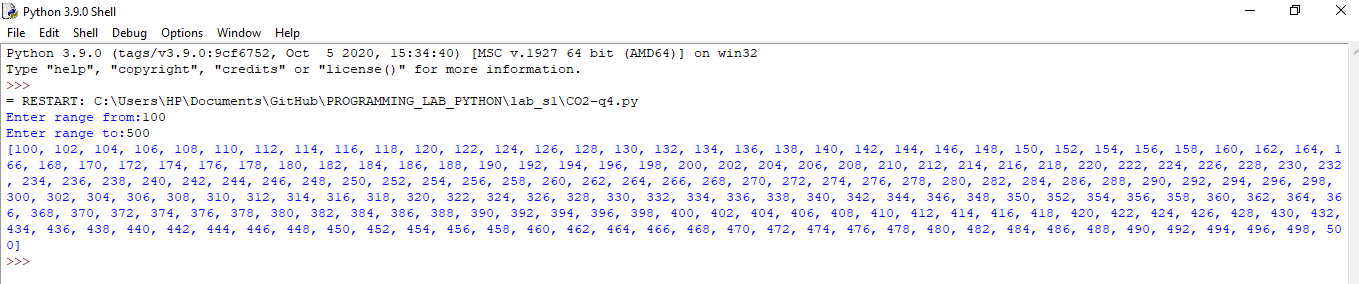
if x%2==0 and x\*\*2:

a.append(x)

print(a)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 5**

**AIM:** Display the given pyramid with step number accepted from user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

**ALGORITHM**

**PROGRAM:**

rows =int(input("Enter the number of rows: "))

for i in range(1,rows+1):

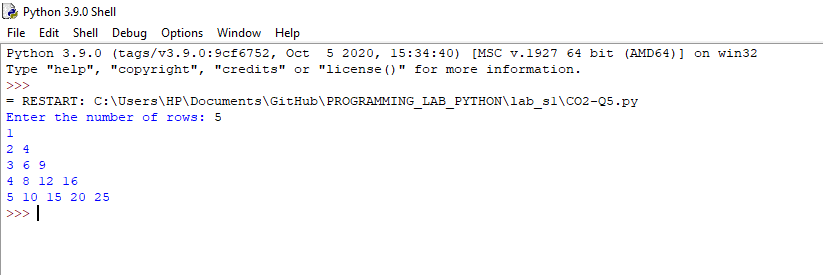
for j in range( 1,i+1 ):

print(i\*j, end=' ')

print(" ")

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 6**

**AIM:** Count the number of characters (character frequency) in a string.

**ALGORITHM**

**PROGRAM:**

st=str(input("ENTER STRING:"))

count=0

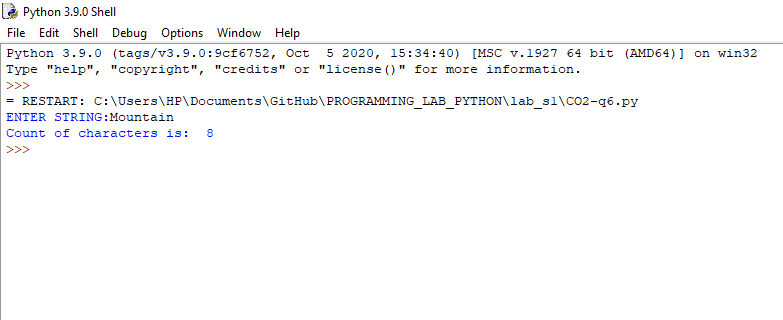
for i in st:

count=count+1

print("Count of characters is: " ,count)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 7**

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

**ALGORITHM**

**PROGRAM:**

s=str(input("ENTER STRING:"))

if s[-3:] == "ing":

s += "ly"

print(s)

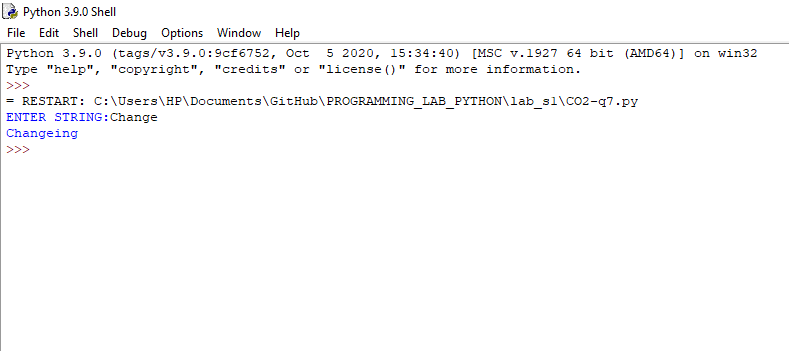
else:

s+="ing"

print(s)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 8**

**AIM:** Accept a list of words and return length of longest word.

**ALGORITHM**

**PROGRAM:**

a = lst.split(",")

def len\_log(list1):

max=len(list1[0])

for i in list1:

if len(i)>max:

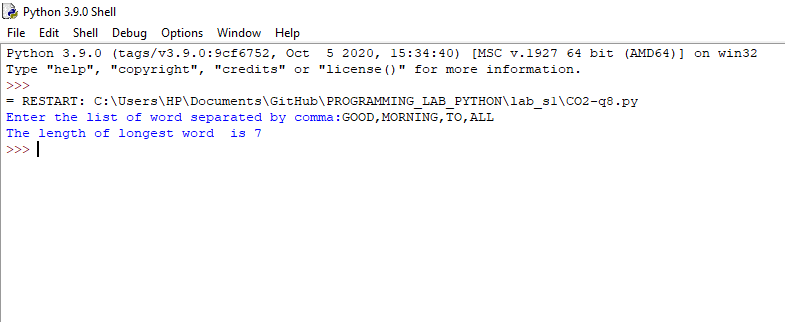
max=len(i)

return max

print("The length of longest word is",len\_log(a))

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 9**

**AIM:** Construct following pattern using nested loop

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**ALGORITHM**

**PROGRAM:**

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 second pattern

for i in range(rows , 0, -1):

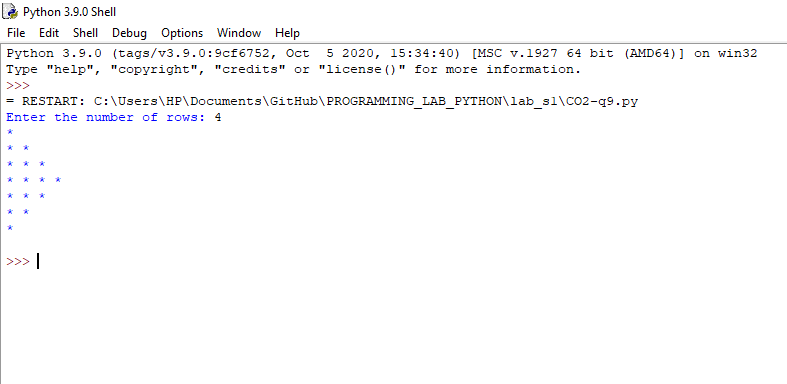
for j in range(0, i - 1):

print("\*", end=' ')

print(" ")

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 10**

**AIM:** Generate all factors of a number.

**ALGORITHM**

**PROGRAM:**

N=int(input("ENTER A NUMBER:"))

print("The factors of {} are,".format(N))

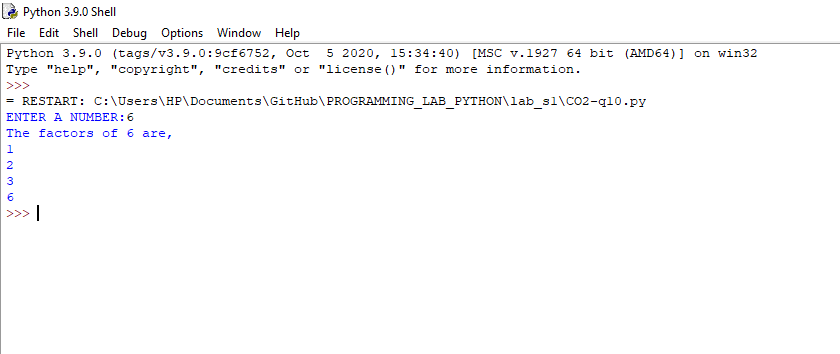
for i in range(1,N+1):

if N % i == 0:

print(i)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 11**

**AIM:** Write lambda functions to find area of square, rectangle and triangle

**ALGORITHM**

**PROGRAM:**

S=int(input("ENTER SIDE:"))

l=int(input("ENTER LENGTH:"))

b=int(input("ENTER BREADTH:"))

h=int(input("ENTER HEIGHT:"))

x = lambda a : a \* a

print("AREA OF SQUARE IS ",(x(S)))

y = lambda a, b : a \* b

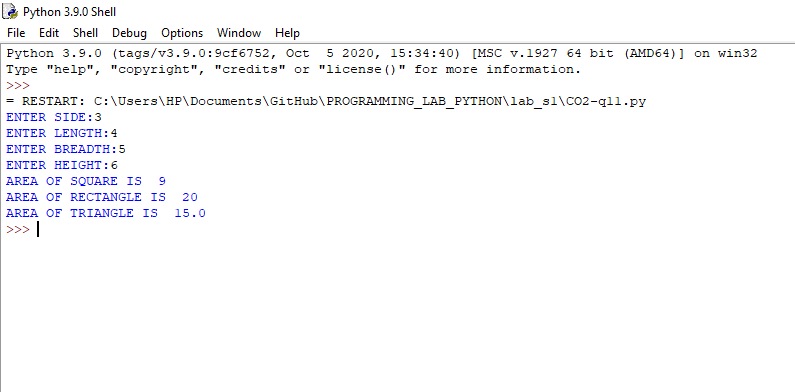
print("AREA OF RECTANGLE IS ",(y(l, b)))

z = lambda a, b : 1/2 \* (a \* b)

print("AREA OF TRIANGLE IS ",(z(b, h)))

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**COURSE OUTCOME 3**

**PROGRAM NO: 1**

**AIM:** Work with built-in packages

**ALGORITHM**

**PROGRAM:**

import platform

x = platform.system()

print(x)

print(end="\n")

import platform

x = dir(platform)

print(x)

print(end="\n")

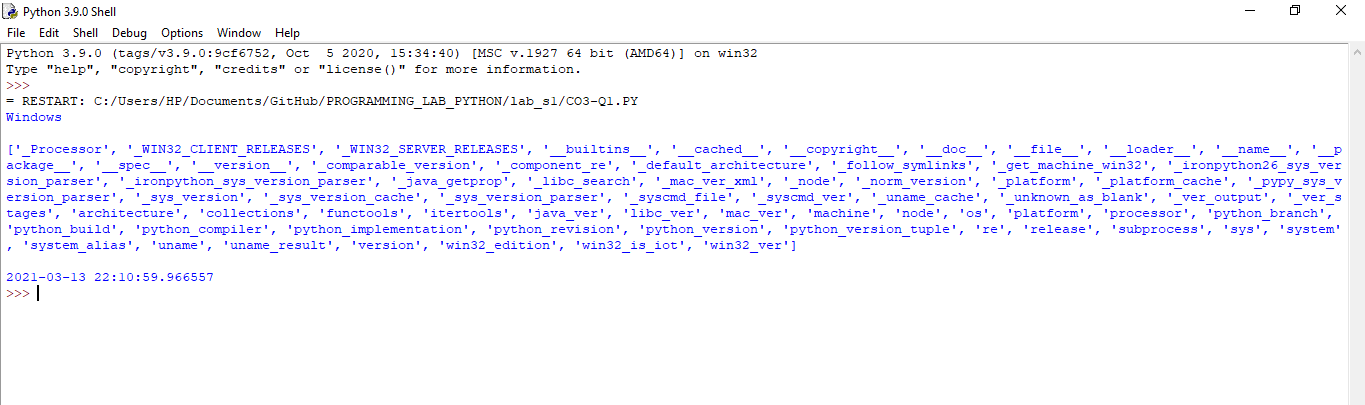
import datetime

x = datetime.datetime.now()

print(x)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 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**

**PROGRAM:**

Graphics

* Graphics3D
* \_\_init\_\_.py
* Cuboid.py
* Sphere.py
* \_\_init\_\_.py
* Circle.py
* Rectangle.py

pkg.py

Circle.py

from math import pi

def area(r):

return pi\*r\*r

def perimeter(r):

return 2\*pi\*r

Rectangle.py

def Rarea(l,b):

return l\*b

def Rperimeter(l,b):

return (2\*(l+b))

Cuboid.py

def CUarea(l,w,h):

return 2\*(l\*w+l\*h+h\*w)

def CUperimeter(l,w,h):

return 4\*(l+w+h)

Sphere.py

from math import pi

def Sarea(r):

return 4\*pi\*r\*r

def Sperimeter(r):

return ((4/3)\*pi\*r\*r\*r)

pkg.py

from Graphics.Circle import \*

from Graphics.Rectangle import \*

from Graphics.Graphics3D.Cuboid import \*

from Graphics.Graphics3D.Sphere import \*

print("CIRCLE")

r=int(input("Enter radius of circle:"))

print("AREA OF CIRCLE:",area(r))

print("AREA OF CIRCLE:",perimeter(r))

print("\nRECTANGLE")

l=int(input("Enter length:"))

b=int(input("Enter breadth:"))

print("AREA OF RECTANGLE:",Rarea(l,b))

print("AREA OF RECTANGLE:",Rperimeter(l,b))

print("\nCUBOID")

l=int(input("Enter length:"))

w=int(input("Enter width:"))

h=int(input("Enter height:"))

print("AREA OF CUBOID:",CUarea(l,w,h))

print("AREA OF CUBOID:",CUperimeter(l,w,h))

print("\nSPHERE")

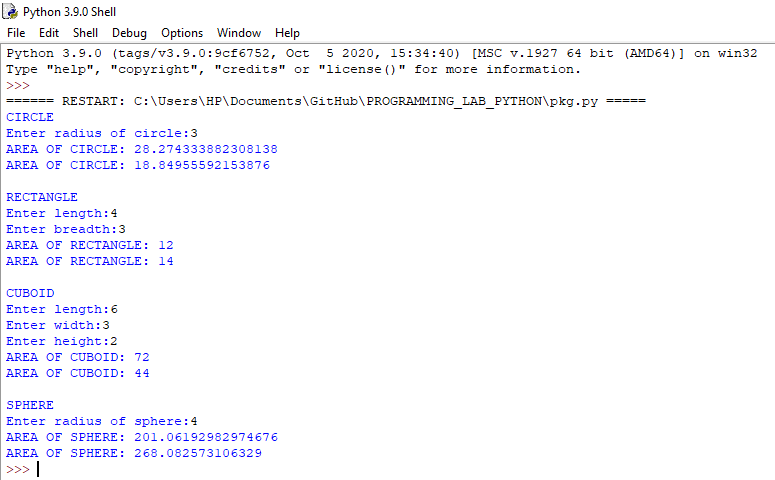
r=int(input("Enter radius of sphere:"))

print("AREA OF SPHERE:",Sarea(r))

print("AREA OF SPHERE:",Sperimeter(r))

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**COURSE OUTCOME 4**

**PROGRAM NO: 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**

**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)

def compare(R1,R2):

if R1.area()>R2.area():

print("\n Rectangles 1 is bigger")

else:

print("\nRectangle 2 is bigger")

return

a=int(input("Enter length of 1st rectangle:"))

b=int(input("Enter length of 1st rectangle:"))

c=int(input("Enter length of 2nd rectangle:"))

d=int(input("Enter length of 2nd rectangle:"))

R1=Rectangle(a,b)

R2=Rectangle(c,d)

print("\nArea of 1st Rectangle :",R1.area())

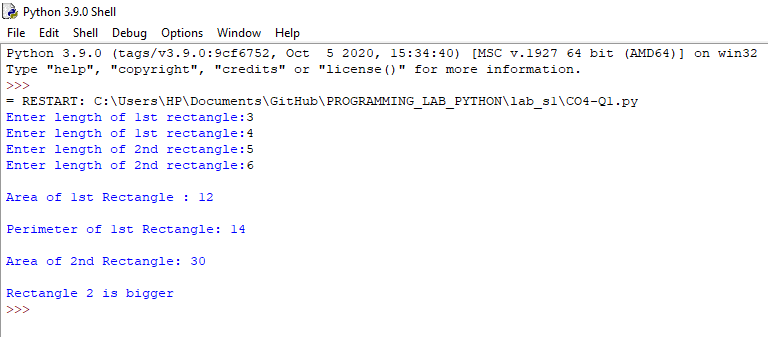
print("\nPerimeter of 1st Rectangle:",R1.perimeter())

print("\nArea of 2nd Rectangle:",R2.area())

R1.compare(R2)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 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**

**PROGRAM:**

class Bankaccount:

def \_\_init\_\_(self):

accno=float(input("Enter Account Number: "))

name=str(input("Enter name of Account Holder:"))

typeofacc=str(input("Enter Type of Account:"))

self.balance=0

def deposit(self):

amount = float(input("\nEnter amount to be deposited: "))

self.balance += amount

print("\n Amount Deposited:", amount)

def withdraw(self):

amount = float(input("\nEnter amount to be withdrawn: "))

if self.balance >= amount:

self.balance -= amount

print("\n You Withdrew:", amount)

else:

print("\n Insufficient balance ")

def display(self):

print("\nAvailable balance is:",self.balance)

s = Bankaccount()

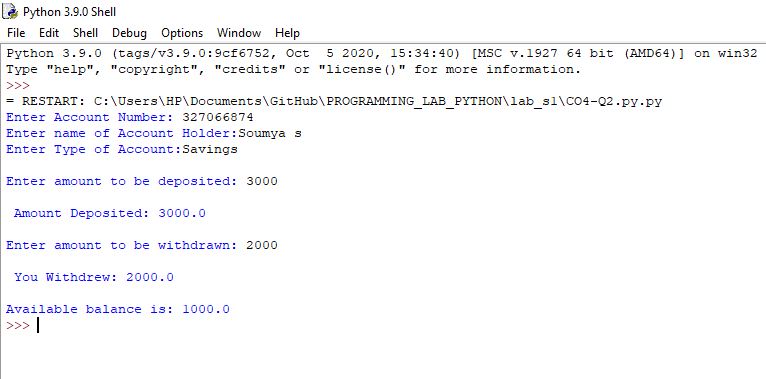
s.deposit()

s.withdraw()

s.display()

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 3**

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

**ALGORITHM**

**PROGRAM:**

class Rectangle:

\_\_length=0

\_\_width=0

\_\_area=0

def \_\_init\_\_(self,l,w):

self.\_\_length=l;

self.\_\_width=w;

def area(self):

return self.\_\_length\*self.\_\_width

def \_\_lt\_\_(self,other):

if(self.\_\_area<other.\_\_area):

return True

else:

return False

obj1=Rectangle(2,3)

obj2=Rectangle(1,2)

print("Area of 1st Rect:",obj1.area())

print("Area of 2nd Rect",obj2.area())

if(obj1.area()<obj2.area()):

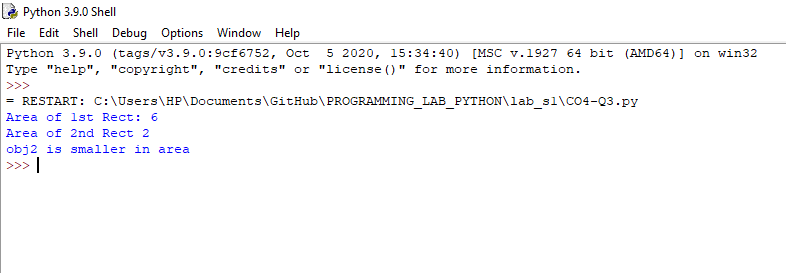
print("obj1 is smaller in area")

else:

print("obj2 is smaller in area")

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 4**

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

**ALGORITHM**

**PROGRAM**

class Time:

def \_\_init\_\_(self,h,m,s):

self.\_\_hour=h

self.\_\_minute=m

self.\_\_second=s

def time(self):

if self.\_\_second>=60:

self.\_\_second-=60

self.\_\_minute+=1

if self.\_\_minute>=60:

self.\_\_minute-=60

self.\_\_hour+=1

return("%.2d:%.2d:%.2d"%(self.\_\_hour,self.\_\_minute,self.\_\_second))

def \_\_add\_\_(self,other):

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

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

\_\_second=self.\_\_second+other.\_\_second

return("%.2d:%.2d:%.2d"%(\_\_hour,\_\_minute,\_\_second))

t1=Time(2,60,60)

print("TIME 1",t1.time())

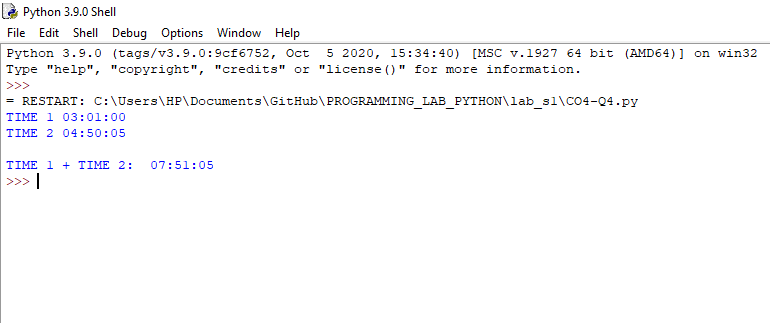
t2=Time(4,50,5)

print("TIME 2",t2.time())

print("\nTIME 1 + TIME 2: ",(t1 + t2))

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO: 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**

**PROGRAM**

class Publisher:

def \_\_init\_\_(self,Pubname):

self.Pubname=Pubname

def display(self):

print("Publisher name is:",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)

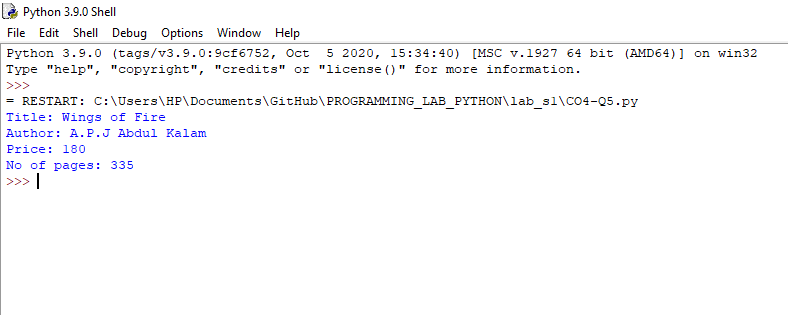
print("No of pages:",self.no\_of\_pages)

b1=Python("DC BOOKS","Wings of Fire","A.P.J Abdul Kalam", 180, 335)

b1.display()

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**COURSE OUTCOME 5**

**PROGRAM NO : 1**

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

**ALGORITHM**

**PROGRAM:**

st1="Good Morning""\n""Have a Nice Day""\n""Are you okay ?""\n"

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

fw.write(st1)

fw.close()

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

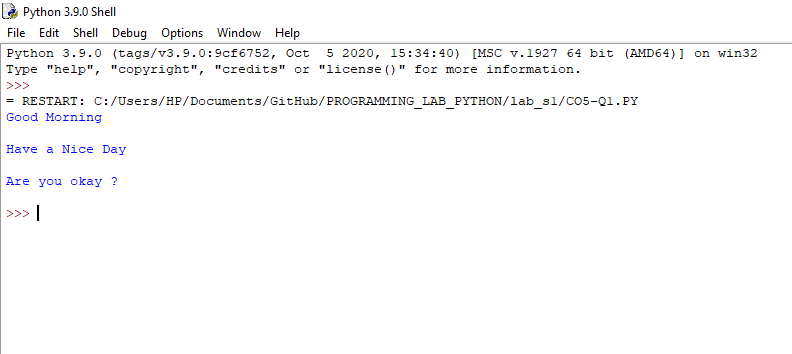
st2=fr.readlines()

for i in st2:

print(i)

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO : 2**

**AIM:** Python program to copy odd lines of one file to other

**ALGORITHM:**

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

str1 = f.readlines()

f.close()

f = open("sample2.txt",'w')

x = 0;

for i in str1:

x = x+1

if x%2!=0:

f.write(i)

f.close()

f=open("sample2.txt",'r')

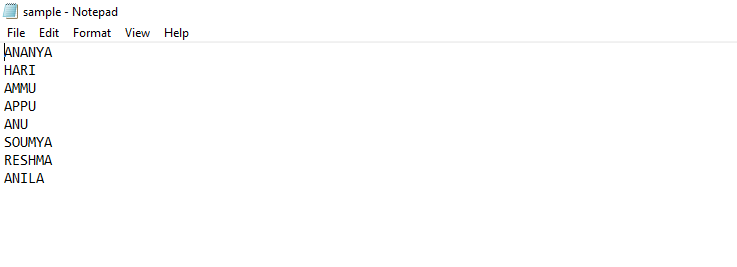
str2=f.readlines()

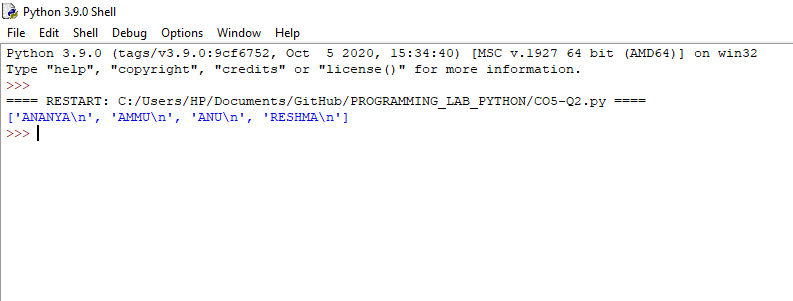
print(str2)

**PROGRAM**

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**





**PROGRAM NO : 3**

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

**ALGORITHM**

**PROGRAM:**

import csv

with open('movie1.csv','w',newline='')as file:

writer=csv.writer(file)

writer.writerow(["SN","Movie","Rating"])

writer.writerow([1,"Lord of the Rings",5])

writer.writerow([2,"Harrypotter",6])

with open('movie1.csv')as csvfile:

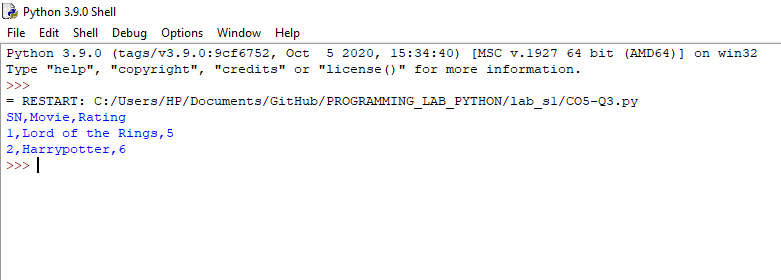
data=csv.reader(csvfile)

for row in data:

print(','.join(row))

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**



**PROGRAM NO : 4**

**AIM:** Write a Python program to read specific columns of a given CSV file and print the content of the columns.

**ALGORITHM**

**PROGRAM**

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**

**PROGRAM NO : 5**

**AIM:** Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

**ALGORITHM**

**PROGRAM:**

import csv

f=open("fruits.csv","w")

writer=csv.DictWriter(f,fieldnames=["fruit","count"])

writer.writeheader()#writeheader() write headers to the csvfile

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 reader:

if c==0:

print(f'{" ".join(row)}')

print(f'{row["fruit"]},{row["count"]}')

f.close()

**RESULT:** The above program is successfully executed and obtained the output

**OUTPUT**

