**FEDERAL INSTITUTE OF**

**SCIENCE AND TECHNOLOGY**

**(FISAT)TM**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



‘**FOCUS ON EXCELLENCE’**

**LABORATORY RECORD**

**20MCA131 - PROGRAMMING LAB**

**Name: ATHIRA RAMACHANDRAN**

**Branch: MASTER OF COMPUTER APPLICATIONS**

**Semester: 1 Batch: A Roll No: 42**

**MARCH 2022**

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(FISAT)TM**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



‘**FOCUS ON EXCELLENCE’**

**CERTIFICATE**

This is to certify that this is the Bonafide record of the Practical work donebyMs. **ATHIRA RAMACHANDRAN** in the **20MCA131 -** **PROGRAMMING LAB** Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.

Signature of Staff in Charge Signature of H.O.D

Name: Name:

Date:

**Date of University practical examination ………………………**

Signature of Signature of

Internal Examiner External Examiner

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI No:** | **Date:** | **Name of Experiment:** | **Page No:** | **Signature of**  **Staff –In –Charge:** |
| **1** |  | Display future leap years from current year to a final year entered by user. | **7** |  |
| **2** |  | 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) | **7** |  |
| **3** |  | Count the occurrences of each word in a line of text. | **9** |  |
| **4** |  | Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead. | **10** |  |
| **5** |  | Store a list of first names. Count the occurrences of ‘a’ within the list | **11** |  |
| **6** |  | 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 | **12** |  |
| **7** |  | Get a string from an input string where all occurrences of first character replaced with  ‘$’, except first character.  [eg: onion -> oni$n] | **14** |  |
| **8** |  | Create a string from given string where first and last characters exchanged. [eg: python -  > nythop] | **15** |  |
| **9** |  | Accept the radius from user and find area of circle. | **15** |  |
| **10** |  | Find biggest of 3 numbers entered. | **16** |  |
| **11** |  | Accept a file name from user and print extension of that. | **17** |  |
| **12** |  | Create a list of colors from comma-separated color names entered by user. Display  first and last colors. | **17** |  |
| **13** |  | Accept an integer n and compute n+nn+nnn. | **18** |  |
| **14** |  | Print out all colors from color-list1 not contained in color-list2. | **19** |  |
| **15** |  | Create a single string separated with space from two strings by swapping the character at position 1. | **20** |  |
| **16** |  | Sort dictionary in ascending and descending order. | **20** |  |
| **17** |  | Merge two dictionaries. | **21** |  |
| **18** |  | Find gcd of 2 numbers. | **22** |  |
| **19** |  | From a list of integers, create a list removing even numbers. | **21** |  |
| **20** |  | Program to find the factorial of a number | **23** |  |
| **21** |  | Generate Fibonacci series of N terms | **24** |  |
| **22** |  | Find the sum of all items in a list | **25** |  |
| **23** |  | Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square. | **26** |  |
| **24** |  | Display the given pyramid with step number accepted from user. Eg: N=4  1  2 4  3 6 9  4 8 12 16 | **27** |  |
| **25** |  | Count the number of characters (character frequency) in a string. | **28** |  |
| **26** |  | Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’ | **30** |  |
| **27** |  | Accept a list of words and return length of longest word. | **31** |  |
| **28** |  | Construct following pattern using nested loop  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* | **32** |  |
| **29** |  | Generate all factors of a number. | **33** |  |
| **30** |  | 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) | **34** |  |
| **31** |  | Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area. | **39** |  |
| **32** |  | 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. 3. Create a class Rectangle with private attributes length and width. Overload ‘ | **41** |  |
| **33** |  | Create a class Rectangle with private attributes length and width. Overload ‘ | **43** |  |
| **34** |  | Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time. | **45** |  |
| **35** |  | 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. | **46** |  |
| **36** |  | Write a Python program to read a file line by line and store it into a list. | **48** |  |
| **37** |  | Write a Python program to read specific columns of a given CSV file and print the content of the columns. | **49** |  |

**C01**

**Experiment Number:1**

**Aim:**

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

**Program Code:**

currentyear=int(input("enter the current year"))

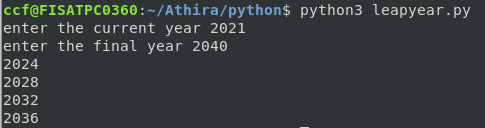
finalyear=int(input("enter the final year"))

for year in range(currentyear,finalyear):

if(year%400==0)or(year%100!=0)and(year%4==0):

print(year)

**output:**



**Experiment Number:2**

**Aim:**

2)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 period

d)List ordinal value of each element of a word

**Program code:**

print("elements in the list are")

list1=[11,-24,0,27,-30]

for num in list1:

if num>=0:

print(num)

print("elements in the list are")

list1=[3,6,9,12,15]

for s in list1:

num=s\*s

print(num)

l=[]

word=input("enter a word")

for i in word:

if i in "aAeEiIOoUu":

l.append(i)

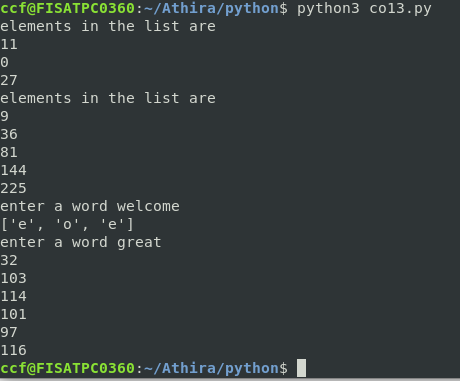
print(l)

word=input("enter a word")

for i in word:

print(ord(i))

**output:**



**Experiment Number:3**



3)Count the occurences of each word in a line of text.

**Program Code:**

list1=[]

list2=[]

x=input("Enter a string:")

for i in x.split(" "):

list1.append(i)

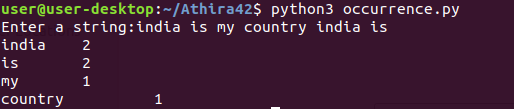
if i not in list2:

list2.append(i)

for i in list2:

print(i,"\t",list1.count(i))

**output:**



**Experiment Number:4**



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

**Program Code:**

list=[]

print("enter 4 integer numbers")

for i in range(4):

j=int(input())

if j>100:

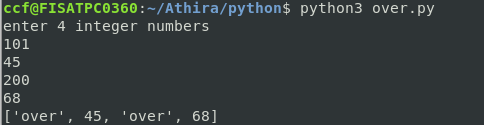
list.append('over')

else:

list.append(j)

print (list)

**output:**



**Experiment Number:5**



5)Store a list of first names.count the occurences of ‘a’ within the list

**Program Code:**

list=[]

l=[]

print("enter 5 names")

for i in range(5):

list.append(input())

for i in list:

count=0

for j in i:

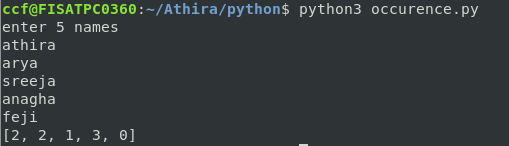
if(j=='a'):

count=count+1

l.append(count)

print(l)

**output**:



**Experiment Number:6**



6)Enter 2 list of integers

a)whether list are of same length

b)whether list sums to same value

c)whether any value occur in both

**Program code:**

l1=[2,6,3,9,]

l2=[4,5,6]

x=len(l1)

y=len(l2)

if x==y:

print("list are of same length")

else:

print("list are of different length")

s1=0

s2=0

for i in range(x):

s1=s1+l1[i]

print("the sum of 1st list:",s1)

for j in range(y):

s2=s2+l2[j]

print("The sum of second list:",s2)

if s1==s2:

print("sum of list are same")

else:

print("sum of list are different")

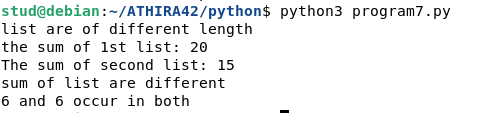
for i in range (x):

for j in range (y):

if l1[i]==l2[j]:

print(l1[i], "and" ,l2[j],"occur in both")

**output**:



**Experiment Number:7**



7)Get a string from an input string where all occurences of first character replaced with ‘$’,except first character.

**Program Code:**

str1=input("enter a string ")

print("original string:",str1)

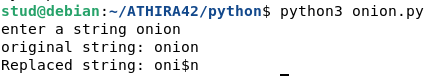
char=str1[0]

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

str1=char+str1[1:]

print("Replaced string:",str1)

**output:**



**Experiment Number:8**



8)create a string from given string where first and last characters exchanged.

**Program Code:**

s="python"

t=s[0]

t1=s[-1]

n=len(s)

ns=t1+s[1:n-1]+t

print(ns)

**output:**



**Experiment Number:9**



9) Accept the radius from user and find area of circle

**Program Code:**

p=int(input("enter the radius- "))

a=3.14\*p\*p

print("Area of Circle= ",a)

**output:**



**Experiment Number:10**



10) Find biggest of 3 numbers entered

**Program Code:**

a=int(input("enter the first number"))

b=int(input("enter the second number"))

c=int(input("enter the third number"))

if a>b:

if a>c:

print(a)

else:

print(c)

else:

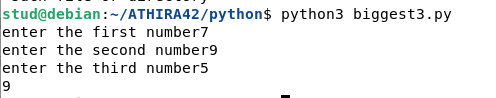
if b>c:

print(b)

else:

print(c)

**output:**



**Experiment Number:11**



11)Accept a file name from user and print extension of that.

**Program Code:**

import os

a=input("Enter the file name:")

print("The extension of file",a,"is",os.path.splitext(a))

**output:**



**Experiment Number:12**



12)create a list of colors from cpmma separated color names entered by user. Display first and last colors

**Program Code:**

color=[]

color=[i for i in input("enter the color:").split(',')]

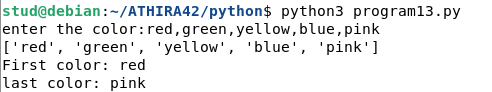
print(color)

i=len(color)-1

print("First color:",color[0])

print("last color:",color[i])

**output:**



**Experiment Number:13**



13)Accept an integer n and compute n+nn+nnn

**Program Code:**

x=int(input("Enter an integer"))

n1=str(x)

n2=n1+n1

n3=n2+n1

result=int(n1)+int(n2)+int(n3)

print(result)

**output:**



**Experiment Number:14**



14)print out of all colors from color-list1 not contained in color-list2

**Program Code:**

l1=['pink','red','black','green','blue']

l2=['pink','yellow','red','violet']

l3=[]

flag=0

print('colours not in l2 are')

for i in l1:

if i not in l2:

l3.append(i)

flag=1

print(l3)

if flag==0:

print('not present')

**output:**



**Experiment Number:15**



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

**Program Code:**

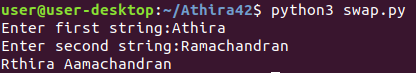
str1=input("Enter first string:")

str2=input("Enter second string:")

str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]

print(str3)

**output:**



**Experiment Number:16**



16)Sort dictionary in ascending and descending order.

**Program Code:**

dict1={"a":1,"c":3,"d":2,"b":4}

l=list(dict1.items())

print(l)

l.sort()

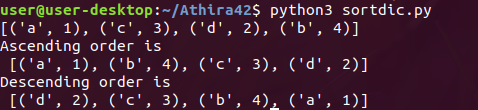
print("Ascending order is\n",l)

l=list(dict1.items())

l.sort(reverse=True)

print("Descending order is\n",l)

**output:**



**Experiment Number:17**



17)Merge two dictionaries.

**Program Code:**

dict1={"Name":"Anu","Age":40}

dict2={"Gender":"F","Qualification":"PG"}

dict1.update(dict2)

print(dict1)

**output:**



**Experiment Number:18**



18)Find gcd of 2 numbers

**Program Code:**

x=int(input("enter the first number"))

y=int(input("enter the second number"))

if x>y:

smallest=y

else:

smallest=x

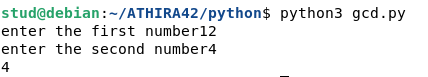
for i in range(1,smallest+1):

if((x%i==0)and(y%i==0)):

hcf=i

print(hcf)

**output:**



**Experiment Number:19**



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

**Program Code:**

l1=[1,2,3,4,5,6]

l2=[]

for i in l1:

if i%2!=0:

l2.append(i)

print(l2)

**output:**



**C02**

**Experiment Number:20**

**Aim:**

20)Program to find the factorial of a number.

**Program Code:**

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

fact=1

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

fact=fact\*i

print(fact)

**output:**

**Experiment Number:21**



21)Generate Fibonacci series of N terms.

**Program Code:**

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

f1=0

f2=1

print(f1)

print(f2)

for i in range(0,n):

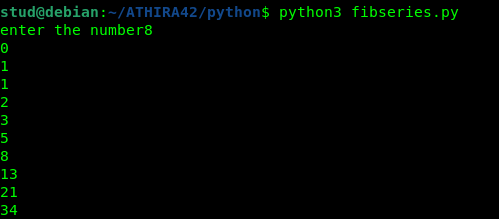
f3=f1+f2

print(f3)

f1=f2

f2=f3

**output:**



**Experiment Number:22**



22)Find the sum of all items in a list.

**Program Code:**

list1=[1,2,3,4,5]

sum=0

for i in list1:

sum=sum+i

print(sum)

**output:**



**Experiment Number:23**



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

**Program Code:**

limit1=1000

limit2=9999

list1=[]

for i in range(limit1,limit2):

j=i

digit=[]

while(i!=0):

digit.append(i%10)

i=int(i/10)

count=0

for n in digit:

if n%2==0:

count=count+1

if count==4:

for k in range(31,100):

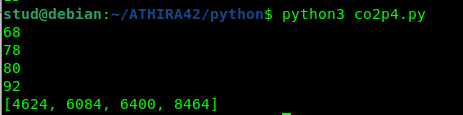
if((k\*\*2)==j):

list1.append(j)

print(k)

print(list1)

**output:**



**Experiment Number:24**



24)Display the given pyramid with step number accepted from user.

EG:N=4

1

2 4

3 6 9

4 8 12 16

**Program code:**

n=int(input("Enter a number:"))

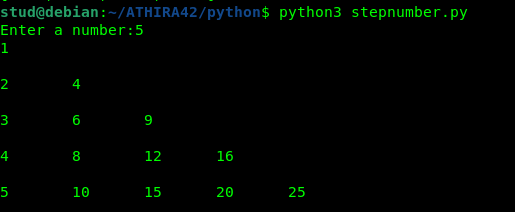
for i in range(1,n+1):

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

print(j,"\t",end="")

print("\n")

**output:**



**Experiment Number:25**



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

**Program Code:**

string=input("Enter a string:")

list1=[]

for i in string:

if i not in list1:

list1.append(i)

for i in list1:

count=0

for j in string:

if(i==j):

count=count+1

print(i,"\t:",count)

**output:**



**Experiment Number:26**



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

**Program code:**

string=input("Enter a string:")

if(string[-3:]=="ing"):

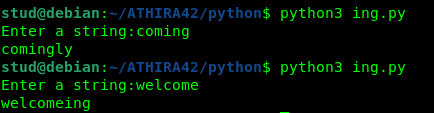
string+="ly"

else:

string+="ing"

print(string)

**output:**



**Experiment Number:27**



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

**Program Code:**

lis=[]

n=int(input("Enter the range:"))

print("Enter the words:")

for i in range(0,n):

lis.append(input(""))

longest=lis[0]

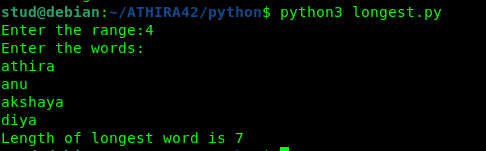
for i in range(1,n):

if(len(lis[i])>len(longest)):

longest=lis[i]

print("Length of longest word is",len(longest))

**output:**



**Experiment Number:28**



28)Construct following pattern using nested loop.

**Code:**

k='\*'

for i in range(1,6):

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

print(k,end="")

print("\n")

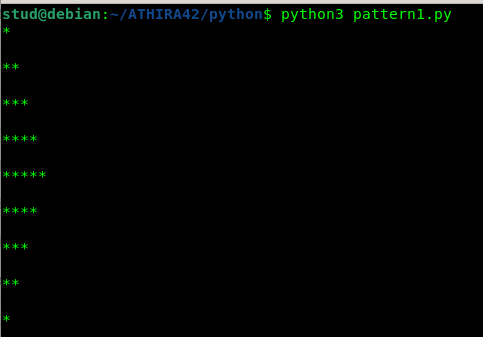
for i in range(5,0,-1):

for j in range(1,i):

print(k,end="")

print("\n")

**output:**



**Experiment Number:29**



29)Generate all factors of a number.

**Program Code:**

n=int(input("Enter a number:"))

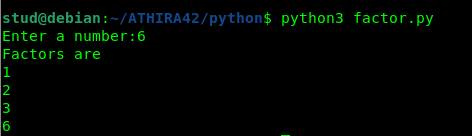
print("Factors are")

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

if(n%i==0):

print(i)

**output:**



**C03**

**Experiment Number:30**

**Aim:**

30)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)

**Program Code:**

**Graphics\circle.py**

from math import pi

def area\_circle(radius):

return pi\*radius\*radius

def perimeter\_circle(radius):

return 2\*pi\*radius

**Graphics\rectangle.py**

def area\_rec(length,width):

return length\*width

def perimeter\_rec(length,width):

return 2\*(length+width)

**Graphics\tdgraphics\cuboid.py**

def area\_cuboid(l,b,h):

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

def volume\_cuboid(l,b,h):

return l\*b\*h

**Graphics\tdgraphics\sphere.py**

from math import pi

def area\_sphere(radius):

return 4\*(pi\*radius\*radius)

def perimeter\_sphere(radius):

return 2\*pi\*radius

**graphics.py (driver code)**

import Graphics

from Graphics import circle,rectangle

from Graphics.tdgraphics import cuboid,sphere

from Graphics.circle import \*

print("Area of a Rectangle with length and width 10 is : ",rectangle.area\_rec(10,10))

print("Perimeter of a Rectangle with length and width 10 is : ",rectangle.perimeter\_rec(10,10))

print("\n")

print("Area of a circle with radius 10 is : ",circle.area\_circle(10))

print("Perimeter of a circle with radius 10 is ",circle.perimeter\_circle(10))

print("\n")

print("Area of a cuboid with length,width,height 10 is : ",cuboid.area\_cuboid(10,10,10))

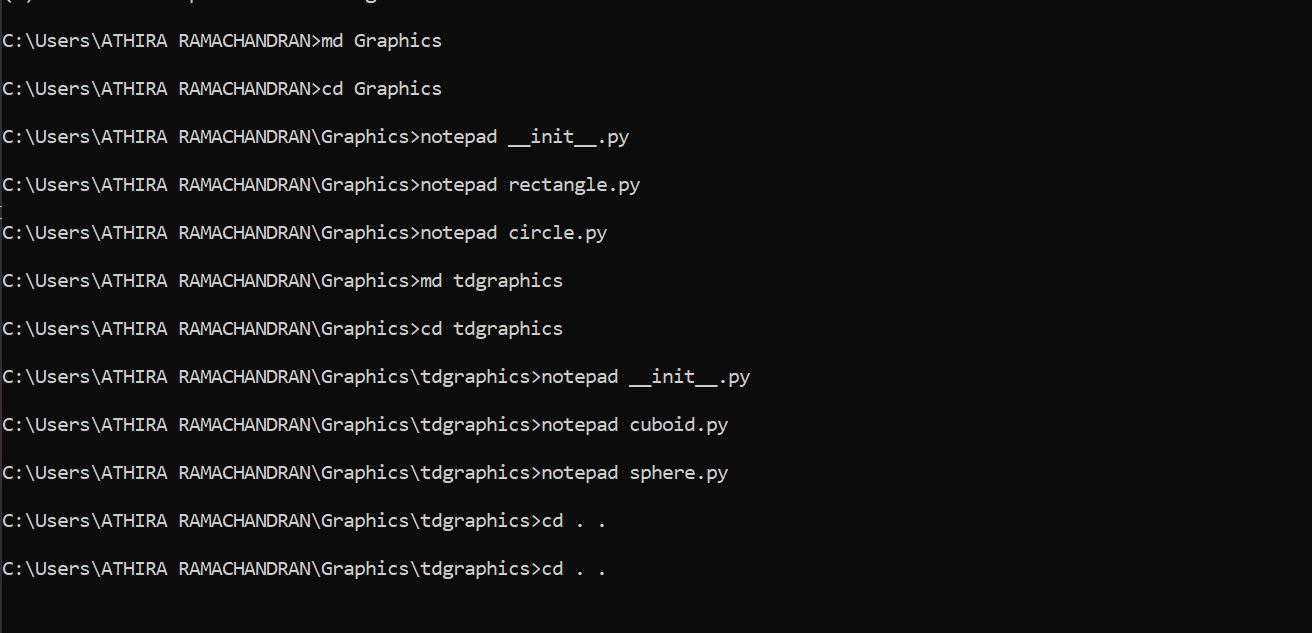
print("Volume of a cuboid with length,width,height 10 is : ",cuboid.volume\_cuboid(10,10,10))

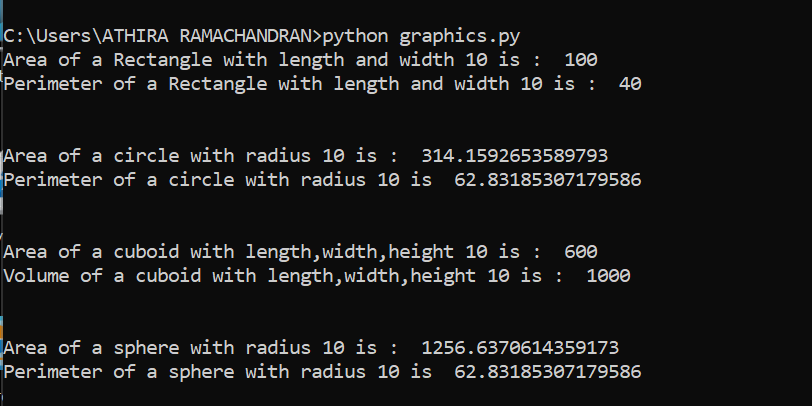
print("\n")

print("Area of a sphere with radius 10 is : ",sphere.area\_sphere(10))

print("Perimeter of a sphere with radius 10 is ",sphere.perimeter\_sphere(10))

**output:**





**C04**

**Experiment Number:31**

**Aim:**

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

**Program Code:**

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)

r1=rectangle(6,4)

r2=rectangle(10,7)

x=r1.area()

y=r2.area()

z=r1.perimeter()

w=r2.perimeter()

print("Area of rectangle1 is",x)

print("Area of rectangle2 is",y)

print("Perimeter of rectangle1 is",z)

print("Perimeter of rectangle2 is",w)

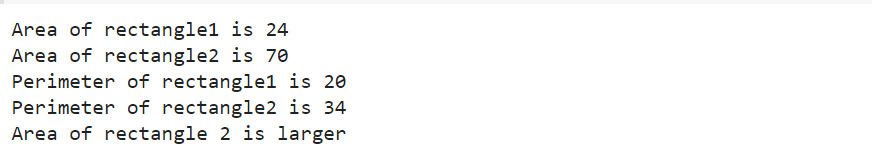
if(x>y):

print("Area of rectangle 1 is larger")

else:

print("Area of rectangle 2 is larger")

**Output:**



**Experiment Number:32**

**Aim:**

32)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 Code:**

class bank:

def \_\_init\_\_(self,acc\_no,name,acc\_type,balance):

self.acc\_no=acc\_no

self.name=name

self.type=acc\_type

self.balance=balance

def withdrawal(self,x):

self.balance=self.balance-x

print("Balance amount after withdrawal:",self.balance)

def deposit(self,y):

self.balance=self.balance+y

print("Balance amount after deposit:",self.balance)

def display(self):

print("Account Number:",self.acc\_no)

print("Account Name:",self.name)

print("Account Type:",self.type)

print("Account Balance:",self.balance)

account1=bank(1234,"Anu","Savings",25000)

account2=bank(3456,"Ammu","Savings",5000)

account3=bank(7890,"Anju","Savings",15000)

account4=bank(4587,"Athira","Savings",4000)

account1.deposit(10000)

account1.withdrawal(2000)

account2.deposit(5000)

account2.withdrawal(1000)

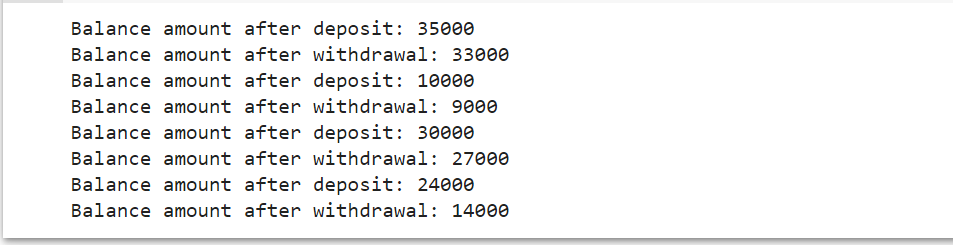
account3.deposit(15000)

account3.withdrawal(3000)

account4.deposit(20000)

account4.withdrawal(10000)

**Output:**



**Experiment Number:33**

**Aim:**

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

**Program Code:**

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 \_\_lt\_\_(self,r2):

if(self.\_\_length\*self.\_\_breadth<r2.\_\_length\*r2.\_\_breadth):

return True

else:

return False

r1=rectangle(10,6)

r2=rectangle(6,4)

x=r1.area()

y=r2.area()

z=r1.perimeter()

w=r2.perimeter()

print("Area of rectangle 1 is",x)

print("Area of rectangle 2 is",y)

print("Perimeter of rectangle 1 is",z)

print("Perimeter of rectangle 2 is",w)

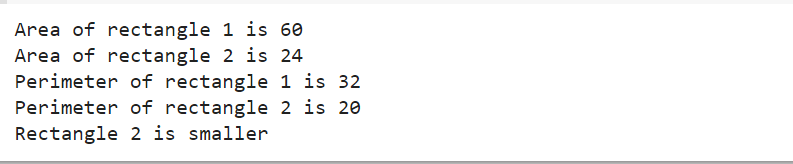
if(r1<r2):

print("Rectangle 1 is smaller")

else:

print("Rectangle 2 is smaller")

**Output:**



**Experiment Number:34**

**Aim:**

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

**Program Code:**

class Time:

def \_\_init\_\_(self,hour,minute,second):

self.\_\_hour=hour

self.\_\_minute=minute

self.\_\_second=second

def \_\_add\_\_(self,t2):

a=self.\_\_hour=t1.\_\_hour+t2.\_\_hour

b=self.\_\_minute=t1.\_\_minute+t2.\_\_minute

c=self.\_\_second=t1.\_\_second+t2.\_\_second

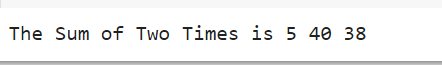
print("The Sum of Two Times is",a,b,c)

t1=Time(1,30,13)

t2=Time(4,10,25)

t3=t1+t2

**Output:**



**Experiment Number:35**

**Aim:**

35)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.

**Program Code:**

class Publisher(object):

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

self.name=name

def display1(self):

print(self.title)

print(self.author)

class Book(Publisher):

def \_\_init\_\_(self,name,title,author):

super().\_\_init\_\_(name)

self.title=title

self.author=author

def display2(self):

super().display1()

print(self.title)

print(self.author)

class Python(Book):

def \_\_init\_\_(self,name,title,author,price,no\_of\_pages):

super().\_\_init\_\_(name,title,author)

self.price=price

self.no\_of\_pages=no\_of\_pages

def display3(self):

super().display2()

print(self.price)

print(self.no\_of\_pages)

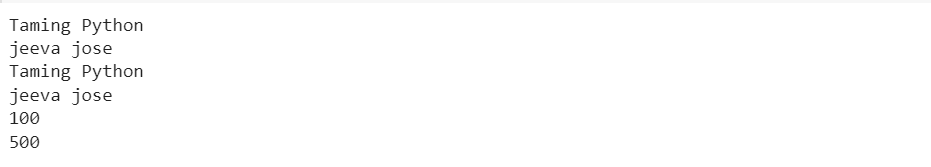
p=Python("ABC Publications","Taming Python","jeeva jose",100,500)

p.display3()

#p.display2()

#p.display3()

**Output:**



**CO5**

**Experiment Number:36**

**Aim:**

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

**Program Code:**

f=open("text\_file.txt",'r')

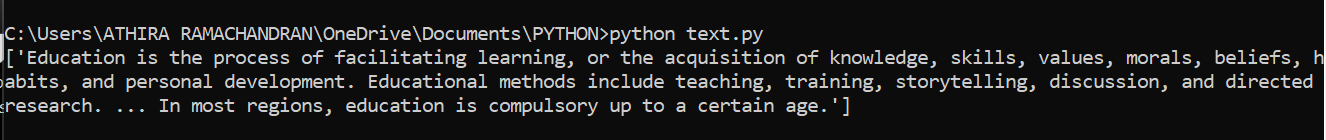
lines=[]

for line in f:

lines.append(line.strip())

print(lines)

**output:**



**Experiment Number:37**

**Aim:**

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

**Program Code:**

import csv

with open('data.csv','r')as file:

reader=csv.reader(file)

for row in reader:

print(row)

**output:**

