**FEDERAL INSTITUTE OF**

**SCIENCE AND TECHNOLOGY**

**(FISAT)TM**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



‘**FOCUS ON EXCELLENCE’**

**LABORATORY RECORD**

**20MCA131 - PROGRAMMING LAB**

**Name : ASWANI K.R**

**Branch : MASTER OF COMPUTER APPLICATION**

**Semester : 1 Batch : 2021 A Roll No : 41**

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(FISAT)TM**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



‘**FOCUS ON EXCELLENCE’**

**CERTIFICATE**

Certified that this is the Bonafede record of the Practical work donebyMrs.**ASWANI K.R** in the **20MCA131- PROGRAMMING** Laboratory of the Federal Institute of Science and Technology 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. |  |  |
| **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) |  |  |
| **3** |  | Count the occurrences of each word in a line of text. |  |  |
| **4** |  | Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead |  |  |
| **5** |  | Store a list of first names. Count the occurrences of ‘a’ within the list |  |  |
| **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. |  |  |
| **7** |  | Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character |  |  |
| **8** |  | Create a string from given string where first and last characters exchanged. [eg: python - > nythop] |  |  |
| **9** |  | Accept the radius from user and find area of circle. |  |  |
| **10** |  | Find biggest of 3 numbers entered. |  |  |
| **11** |  | Accept a file name from user and print extension of that. |  |  |
| **12** |  | Create a list of colors from comma-separated color names entered by user. Display first and last colors. |  |  |
| **13** |  | Accept an integer n and compute n+nn+nnn. |  |  |
| **14** |  | Print out all colors from color-list1 not contained in color-list2. |  |  |
| **15** |  | Create a single string separated with space from two strings by swapping the character at position 1. |  |  |
| **16** |  | Merge two dictionaries. |  |  |
| **17** |  | Find gcd of 2 numbers. |  |  |
| **18** |  | From a list of integers, create a list removing even numbers. |  |  |
| **19** |  | Program to find the factorial of a number |  |  |
| **20** |  | Generate Fibonacci series of N terms |  |  |
| **21** |  | Find the sum of all items in a list |  |  |
| **22** |  | Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square. |  |  |
| **23** |  | Display the given pyramid with step number accepted from user. |  |  |
| **24** |  | Count the number of characters (character frequency) in a string. |  |  |
| **25** |  | Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’. |  |  |
| **26** |  | Accept a list of words and return length of longest word. |  |  |
| **27** |  | Construct following pattern using nested loop.  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |  |  |
| **28** |  | Generate all factors of a number. |  |  |
| **29** |  | 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) |  |  |
| **30** |  | Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area. |  |  |
| **31** |  | 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. |  |  |
| **32** |  | Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles. |  |  |
| **33** |  | Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time |  |  |
| **34** |  | 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. |  |  |
| **35** |  | Write a Python program to read a file line by line and store it into a list. |  |  |
| **36** |  | Write a Python program to read each row from a given csv file and print a list of strings. |  |  |

**COURSE OUTCOME 1**

**PROGRAM 1.1**

**Aim**

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

**SOURCE CODE**

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

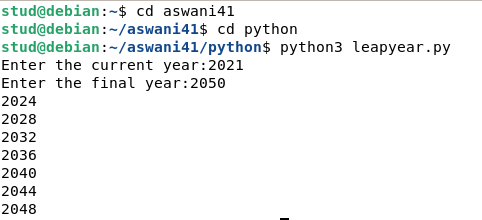
finalyear=int(input("Enter the final year:"))

for year in range(currentyear,finalyear):

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

print(year)

**OUTPUT**



**PROGRAM 1.2**

**Aim**

List comprehensions:

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

(b) Sqare of N numbers.

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

(d) List ordinal value of each element of a word.

**SOURCE CODE**

(a)

print("Elements in the list are")

list1=[2,-7,8,9,0,-3,-1]

for num in list1:

if num>=0:

print(num)

(b)

print("Elements in the list are")

list1=[2,4,7,3,9]

for s in list1:

num=s\*s

print(num)

(c)

L=[]

s=input("Enter a word:")

for i in s:

if i in "AaEeIiOoUu":

L.append(i)

print(L)

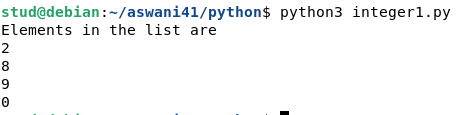
(d)

word=input("Enter a word:\n")

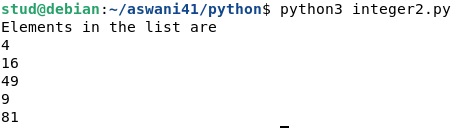
for i in word:

print(ord(i))

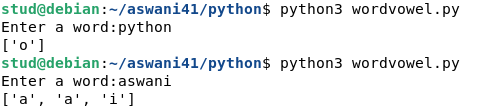
**OUTPUT**

(a)

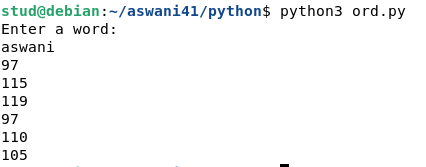
(b)



(c)



(d)



**PROGRAM 1.3**

**Aim**

Count the occurence of each word in a line of text.

**SOURCE CODE**

list1=[]

list2=[]

x=input("Enter a line of text:")

for i in x.split(" "):

list1.append(i)

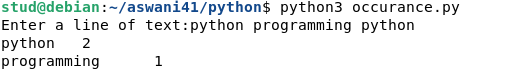
if i not in list2:

list2.append(i)

for i in list2:

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

**OUTPUT**

****

**PROGRAM 1.4**

**Aim**

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

**SOURCE CODE**

list=[]

print("Enter the integer numbers:")

for i in range(4):

j=int(input())

if j>=100:

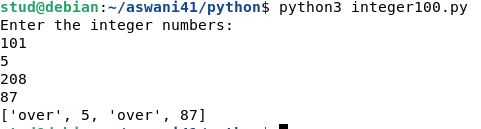
list.append('over')

else:

list.append(j)

print(list)

**OUTPUT**



**PROGRAM 1.5**

**Aim**

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

**SOURCE CODE**

list=[]

l=[]

print("Enter the strings:")

for i in range(3):

list.append(input())

for i in list:

count=0

for j in i:

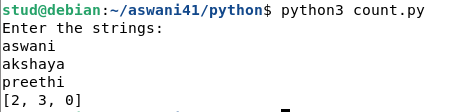
if(j=='a'):

count=count+1

l.append(count)

print(l)

**OUTPUT**



**PROGRAM 1.6**

**Aim**

Enter two list of integers. Check

(a) Whether they are of same length.

(b) Whether list sums to same value.

(c) Whether any value occure in both.

**SOURCE 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")

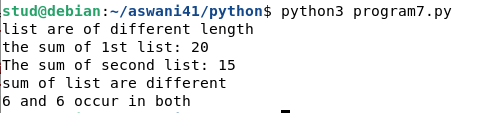
for i in range (x):

for j in range (y):

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

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

**OUTPUT**



**PROGRAM 1.7**

**Aim**

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

[onion -> oni$n]

**SOURCE CODE**

str1=input("Enter the string:")

print("Original string:",str1)

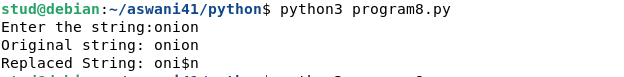
char=str1[0]

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

str1=char+str1[1:]

print("Replaced String:",str1)

**OUTPUT**



**PROGRAM 1.8**

**Aim**

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

[eg : Python ->nythoP]

**SOURCE CODE**

s=input("Enter a string:")

print("Original string",s)

n=len(s)

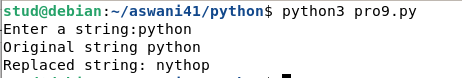
t=s[0]

t1=s[-1]

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

print("Replaced string:",ns);

**OUTPUT**



**PROGRAM 1.9**

**Aim**

Accept the radius from user and find area of circle.

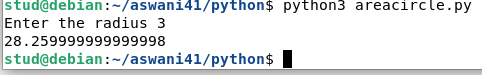
**SOURCE CODE**

p=int(input("Enter the radius"))

a=3.14\*p\*p

print(a)

**OUTPUT**

****

**PROGRAM 1.10**

**Aim**

Find biggest of 3 numbers entered .

**SOURCE CODE**

a=int(input("Enter the 1st number "))

b=int(input("Enter the 2nd number "))

c=int(input("Enter the 3rd number "))

if a>b:

if a>c:

print(a)

else:

print(c)

else:

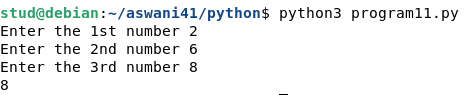
if b>c:

print(b)

else:

print(c)

**OUTPUT**



**PROGRAM 1.11**

**Aim**

Accept a file name from user and print extensin for that.

**SOURCE CODE**

import os

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

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

**OUTPUT**



**PROGRAM 1.12**

**Aim**

Create a list of coloures from comma-seperated colour names entered by user. Display first and last colours.

**SOURCE CODE**

color=[]

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

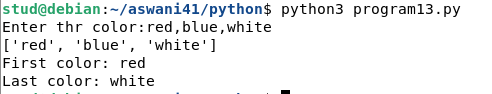
print(color)

i=len(color)-1

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

print("Last color:",color[it

**OUTPUT**



**PROGRAM 1.13**

**Aim**

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

**SOURCE CODE**

i=input("Enter a number:")

j=i+i+i

k=i+i

s=i

s1=int(k)+int(j)+int(s)

print(s1)

**OUTPUT**



**PROGRAM 1.14**

**Aim**

Print out all colours from color list1 not contained in color list2.

**SOURCE CODE**

list1=['red','blue','green','yellow','brown']

list2=['blue','white','black','green','skyblue','ash']

list3=[]

for i in list1:

if i not in list2:

list3.append(i)

print(list3)

**OUTPUT**



**PROGRAM 1.15**

**Aim**

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

**SOURCE CODE**

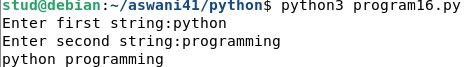
str1=input("Enter first string:")

str2=input("Enter second string:")

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

print(str3)

**OUTPUT**



**PROGRAM 1.16**

**Aim**

Sort dictinary in ascending and descending order.

**SOURCE CODE**

d1={"a":1,"c":3,"d":2,"b":4}

l=list(d1.items())

print(l)

l.sort()

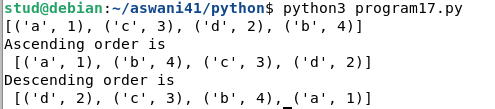
print("Ascending order is\n",l)

l=list(d1.items())

l.sort(reverse=True)

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

**OUTPUT**



**PROGRAM 1.17**

**Aim**

Merge two dictionaries.

**SOURCE CODE**

d1={"name":"Aswani","age":"21"}

d2={"sex":"Female","qualification":"PG"}

d1.update(d2)

print(d1)

**OUTPUT**



**PROGRAM 1.18**

**Aim**

Find gcd of two numbers.

**SOURCE CODE**

x=int(input("Enter the first number:"));

y=int(input("Enter the second number:"));

if x>y:

smallest=y

else:

smallest=x

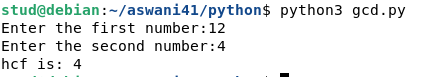
for i in range(1,smallest+1):

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

hcf=i

print("hcf is:",hcf)

**OUTPUT**



**PROGRAM 1.19**

**Aim**

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

**SOURCE CODE**

list1=[1,2,3,4,5,6,7,8,9]

num=[]

for i in list1:

if(i%2!=0):

num.append(i)

print(num)

**OUTPUT**



**COURSE OUTCOME 2**

**PROGRAM 2.1**

**Aim**

Program to find the factorial of a number.

**SOURCE CODE**

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

fact=1

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

fact=fact\*i

print(fact)

**OUTPUT**



**PROGRAM 2.2**

**Aim**

Generate Fibonacci series of N terms.

**SOURCE CODE**

n=int(input("Enter the limit "))

f1=0

f2=1

print(f1)

print(f2)

for i in range(0,n-2):

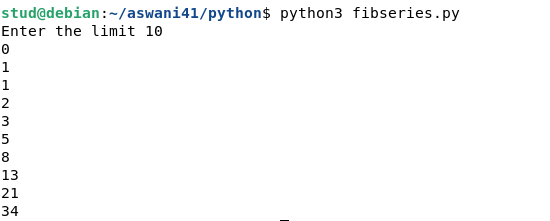
f3=f1+f2

print(f3)

f1=f2

f2=f3

**OUTPUT**

****

**PROGRAM 2.3**

**Aim**

Find the sum of all items in a list.

**SOURCE CODE**

list1=[10,20,40]

sumlist1=0

for i in list1:

sumlist1=sumlist1+i

print("Sum:",sumlist1)

**OUTPUT**



**PROGRAM 2.4**

**Aim**

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

**SOURCE 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):

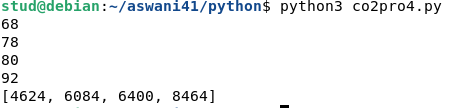
if((k\*\*2)==j):

list1.append(j)

print(k)

print(list1)

**OUTPUT**



**PROGRAM 2.5**

**Aim**

Display the given pyramid with step numbers accepted from user.

Eg : 4

1

2 4

3 6 9

4 8 12 16

**SOURCE CODE**

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

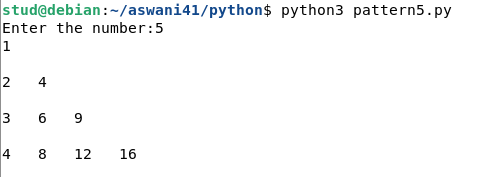
for i in range(1,n):

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

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

print("\n")

**OUTPUT**



**PROGRAM 2.6**

**Aim**

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

**SOURCE 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

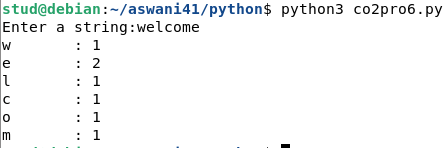
for j in string:

if(i==j):

count=count+1

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

**OUTPUT**



**PROGRAM 2.7**

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

**SOURCE CODE**

string=input("Enter a string:")

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

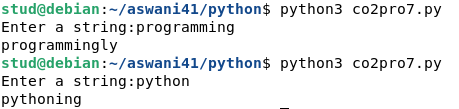
string+="ly"

else:

string+="ing"

print(string)

**OUTPUT**



**PROGRAM 2.8**

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

**SOURCE CODE**

list1=[]

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

print("Enter the words:")

for i in range(0,n):

list1.append(input(""))

longest=list1[0]

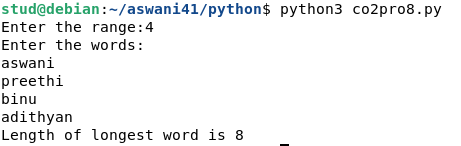
for i in range(1,n):

if(len(list1[i])>len(longest)):

longest=list1[i]

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

**OUTPUT**



**PROGRAM 2.9**

**Aim :** Construct following pattern using nested loop.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**SOURCE CODE**

k='\*'

for i in range(1,6):

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

print(k,end=" ")

print("\n")

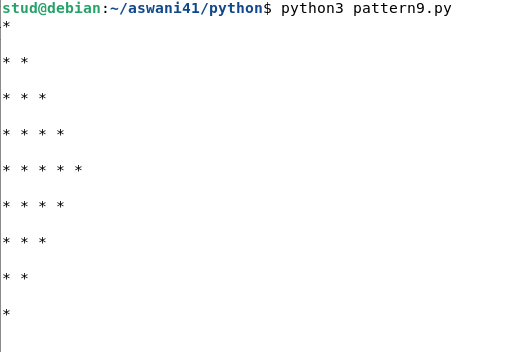
for i in range(5,0,-1):

for j in range(1,i):

print(k,end=" ")

print("\n")

**OUTPUT**



**PROGRAM 2.10**

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

**SOURCE CODE**

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

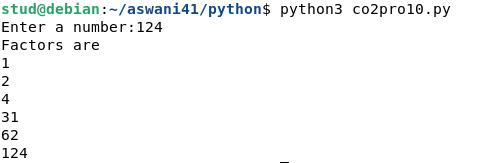
print("Factors are")

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

if(n%i==0):

print(i)

**OUTPUT**



**COURSE OUTCOME 3**

**PROGRAM 3.1**

**Aim :** Create a package graphics with modules rectangle,circle and sub-package 3D graphics with module cuboid and sphere. Include methods to find area and perimeter of respective figures in each modules. Write programs that finds area and perimeter of figures by different importing statements.

**SOURCE 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

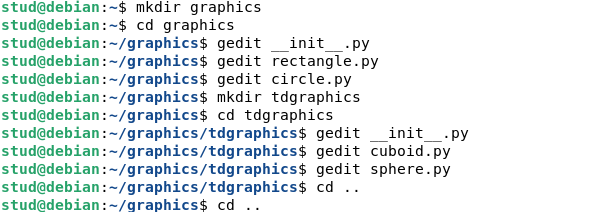
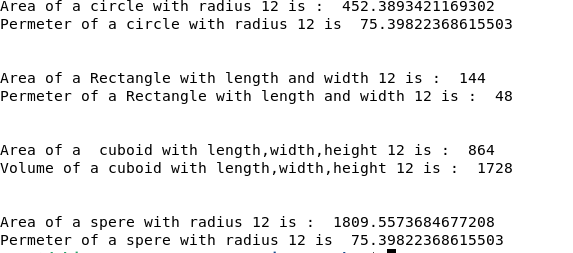
def area\_sphere(radius):

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

def perimeter\_sphere(radius):

return 2\*pi\*radius

**OUTPUT**

****

**COURSE OUTCOME 4**

**PROGRAM 4.1**

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

**SOURCE CODE**

**c**lass Rectangle:

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

self.l=l

self.b=b

def area(self):

return self.l\*self.b

def perimeter(self):

print("Perimeter:",2\*(self.l+self.b))

x=Rectangle(3,6)

y=Rectangle(3,4)

a=x.area()

print("Area of first rectangle:",a)

x.perimeter()

b=y.area()

print("Area of second rectangle:",b)

y.perimeter()

print("Largest area is:")

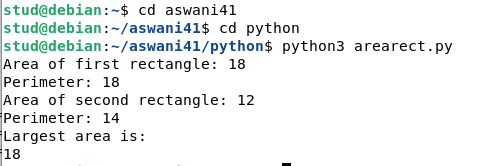
if(a>b):

print(a)

else:

print(b)

**OUTPUT**

****

**PROGRAM 4.2**

**Aim :** Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposite at the bank and withdraw an amount from the bank.

**SOURCE CODE**

class Bank:

def \_\_init\_\_(self,accno,name,acctype,balance):

self.accno=accno

self.name=name

self.acctype=acctype

self.balance=balance

def withdraw(self,x):

if(x<self.balance):

self.balance=self.balance-x

print("Account Number:",self.accno)

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

print("After withdraw:",self.balance)

else:

print("Insufficient Balance")

def deposit(self,y):

self.balance=self.balance+y

print("Account Number:",self.accno)

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

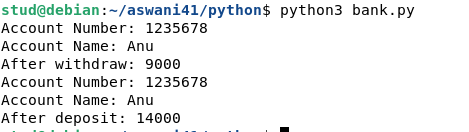
print("After deposit:",self.balance)

ac1=Bank(1235678,"Anu","Savings",10000)

ac1.withdraw(1000)

ac1.deposit(5000)

**OUTPUT**

****

**PROGRAM 4.3**

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

**SOURCE CODE**

class Rectangle:

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

self.length=l

self.breadth=b

def area(self):

return self.length\*self.breadth

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

if((self.length\*self.breadth)<(m.length\*m.breadth)):

return True

else:

return False

r1=Rectangle(8,2)

r2=Rectangle(4,5)

x=r1.area()

y=r2.area()

print("rectangle1 area=",x)

print("rectangle2 area=",y)

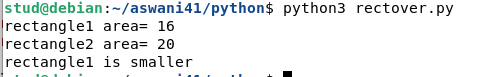
if(r1<r2):

print("rectangle1 is smaller")

else:

print("rectangle2 is smaller")

**OUTPUT**

****

**PROGRAM 4.4**

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

**SOURCE CODE**

class Time:

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

self.\_\_hour=h

self.\_\_minute=m

self.\_\_second=s

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

hour=self.\_\_hour+ob.\_\_hour

minute=self.\_\_minute+ob.\_\_minute

second=self.\_\_second+ob.\_\_second

t=Time(hour,minute,second)

return t

def print\_it(self):

print("Hour :",self.\_\_hour)

print("Minute :",self.\_\_minute)

print("Second :",self.\_\_second)

t1=Time(4,43,10)

t2=Time(8,20,55)

t3=t1+t2

t3.print\_it()

**OUTPUT**

****

**PROGRAM 4.5**

**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 overreading.

**SOURCE CODE**

class Publisher:

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

self.name=name

class Book(Publisher):

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

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

self.title=title

self.auther=auther

def print\_function(self):

print("This Fuction is a member fuction of class Publisher")

class Python(Book):

def \_\_init\_\_(self,name,title,auther,price,nop):

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

self.price=price

self.nop=nop

def print\_function(self):

print("Name :",self.name)

print("Title :",self.title)

print("Auther :",self.auther)

print("Price :",self.price)

print("Number of Pages :",self.nop)

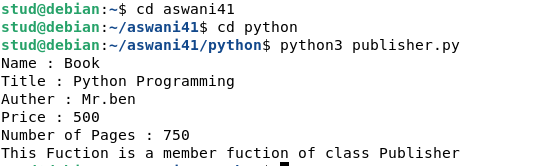
p1=Python("Book","Python Programming","Mr.ben",500,750)

p1.print\_function()

p2=Book("a","b","c")

p2.print\_function()

**OUTPUT**

****

**COURSE OUTCOME 5**

**PROGRAM 5.1**

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

**SOURCE CODE**

file=open("text.txt","r")

lines=[]

for line in file:

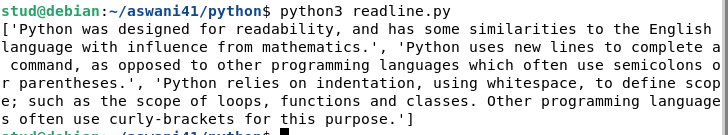
lines.append(line.strip())

print(lines)

**text.txt**

Python was designed for readability, and has some similarities to the English language with influence from mathematics.Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**OUTPUT**

****

**PROGRAM 5.2**

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

**SOURCE CODE**

import csv

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

reader = csv.reader(file)

for row in reader:

print(row)

**peopledetail.csv**

Id,Name,Desig,Salary

1,preethi,Manager,70000

2,Binu,Secretary,35000

003,Adithyan,Security,28000

**OUTPUT**

