Task 1:

employee name and salary , emp ={"ajay":1000,"samy":2000,"jack":3000,"suresh":1000,"ramesh":2900}

#print name and salary of all employee whose salary is greater than 2500 from dictionary

emp ={"ajay":1000,"samy":2000,"jack":3000,"suresh":1000,"ramesh":2900}

filter\_list = {}

for key, value in emp.items():

  if value >= 2500:

   filter\_list[key] = value

   print(filter\_list)

Task 2:

Factorial of number:

def factorial(x):

  if x>0:

    print(x)

    result=factorial(x-1) \* x

    return result

  else:

    return 1

Task 3:

use function, variable length argument (only int - run time) , return value sum of all the arguments

def number(\*n):

  sum=0

  for i in n:

    sum +=i

  return sum

Task 4:

, use lambda function, variable length argument (only int - run time) , return value sum of all the arguments

add = lambda \*num:sum(num)

print( add(5,2,5) )

Task 5:

USER DEFINED PROGRAM

Task

create some emp class -->

Tcs,HCL,Infy

id  , name , salary , location, age -> common

obj --> TCS23 , HCL23, Infy23

Name => malik basha

name -> Firstname,lastname ==> only(HCL)

salary ==> 30000

TCS => 2 floating point (30000.00)

HCL => 10% detect and show (27000)

Infy => Annual (30000\*12=360000)

TCS -> show

location => siruseri  , dob

HCL -> show

Gender, Bloodgroup , doj

Infy->show

Exp, mobile, Address, aadhar

PROGRAM:

class emp:

    def \_\_init\_\_(self,id,name,age,location,salary):

        self.ID=id

        self.NAME=name

        self.AGE=age

        self.LOCATION=location

        self.SALARY=salary

    def show(self):

        print("id is {} , name is {} , age is {} , place is {}".format(self.ID,self.NAME,self.AGE,self.PLACE))

    def modify\_name(self,name):

        self.NAME=name

class HCL(emp):

    def details\_cus(self,name):

       result=self.NAME.split()

       firstname = result[0]

       last\_name=result[1]

       amount=(self.SALARY\*10)/100

       Absolute =self.SALARY-amount

       print("Id is HCL{} , first\_name is {} , last\_name is {} ,salary is {}".format(self.ID,first\_name,last\_name,Absolute))

    def show(self):

      gender=str(input("enter the gender:"))

      Bloodgroup=str(input("enter the Bloodgroup:"))

      doj = str(input("enter the date of joining:"))

      print(("HI I am {} ,bloodgroup is {} ,my date of joining in this company is {} ".format(gender,Bloodgroup,doj)))

class TCS(emp):

  def show(emp):

    location=str(input("enter the location:"))

    dob = str(input("enter the date of birth:"))

    print(("My company location  is {},and  date of birth {}".format(location,dob)))

  def TCS\_salary(self,salary):

    decimalsalary=format(self.SALARY,".2f")

    print("Id is TCS{} ,salary is {}".format(self.ID,decimalsalary))

class Infy(emp):

  def show(self):

       experience=int(input("enter the exp number"))

       mobile =int(input("enter the phone number"))

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

       address=str(input("enter the address"))

       print(("My experience  is {} , mobile is {} , aadhar is {} ,address is {}".format(experience,mobile,aadhar,address)))

  def salary(self,salary):

     annualsalary=self.SALARY \* 12

     print("Id is Infy{} ,salary is {}".format(self.ID,annualsalary))

Task 6:

User defined program to built library

PROGRAM:

class Library:  
 def \_\_init\_\_(self,data):  
 self.booklist=data  
 def show(self):  
 print(self.booklist)  
 def lendBook(self,book): #7self = lib\_salem , book = requestedBook  
 if book in self.booklist: #8a verifies requestedBook in self.booklist , True  
 self.booklist.remove(book) #9 removes requestedBook from self.booklist  
 print("Thank you , Issued book is ", book) #10 acknowledge issued book  
 return True  
 else:#8b  
 print("book is not available") #8b2 acknowledge book is not available  
 return False  
  
class customer:  
 def \_\_init\_\_(self):  
 self.borrowedBookList = []  
 def request\_book(self):#2  
 book = input("enter the book name: ")#3  
 return book #4  
 def addBook(self,book):  
 self.borrowedBookList.append(book)  
 def show(self):  
 print(self.borrowedBookList)  
lib\_salem = Library(["book1","book2","book3","book4","book5","book3"])  
tamil = customer()  
  
while True:  
 print("---------------------------------------------------------------------------------")  
 print("1. show book list")  
 print("2. issue book")  
 print("3. return book")  
 print("4. track customer")  
 print("5. exit")  
 print("---------------------------------------------------------------------------------")  
 choice = int(input("enter your choice: "))  
 if choice == 1:  
 lib\_salem.show()  
 elif choice == 2:  
 requestedBook = tamil.request\_book()#1 #5 retun data from customer is assigned to requestedBook  
 status = lib\_salem.lendBook(requestedBook) #6 passing requestedBook to library for lend purpose #12 sucess for 6  
 if status == True:  
 tamil.addBook(requestedBook)  
 elif choice == 3:  
  
 lib\_salem.booklist.append(requestedBook)  
 tamil.borrowedBookList.remove(requestedBook)  
 print("Thank you ,Visit Again")  
 print("lib:",tamil.show())  
 print("book:",lib\_salem.booklist)  
 elif choice == 4:  
 tamil.show()  
 elif choice == 5:  
 break

Task 7:

Task is in pandas as sql query

#1 identify number of male who has diabetes

len(data[(data['diabetes']==1)&(data['gender']=='Male')].value\_counts())

#2 identify number of person who has bp,suger and heart disease

len(data[(data['hypertension']==1)&(data['blood\_glucose\_level'])&(data['heart\_disease']==1)])

#3 current smoker in both gender

len(data[(data['smoking\_history']=="current")])

#4 identify sugar level>180 and no diabetes

data[(data['blood\_glucose\_level']>180)&(data['diabetes']==0)]

#5 non-adults details who is having sugar

data[(data['age']<18)&(data['blood\_glucose\_level']>200)]

#6 slicing ages into different age slot based on this identify average hba1c and smoking history