COMPUTER SCIENCE WITH PYTHON

PRACTICAL FILE

NAME:

CLASS:

SECTION:

INDEX

	priced less than 50. (items.dat- id,gift,cost).Assume that info is stored in the form of list	
8	A csv file counties.csv contains data in the following order: country,capital,code sample of counties.csv is given below: india,newdelhi,ii us,washington,uu malaysia,ualaumpur,mm france,paris,ff write a python function to read the file counties.csv and display the names of all those countries whose no of characters in the capital are more than 6.	
9	Write a Program in Python that defines and calls the following user defined functions: a) add() – To accept and add data of an employee to a CSV file ‘furdata.csv’. Each record consists of a list with field elements as fid, fname and fprice to store furniture id, furniture name and furniture price respectively. b) search()- To display the records of the furniture whose price is more than 10000.	
10	What is the advantage of using a csv file for permanent storage? Write a Program in Python that defines and calls the following user defined functions: (i) ADD() – To accept and add data of an employee to a CSV file ‘record.csv’. Each record consists of a list with field elements as empid, name and mobile to store employee id, employee name and employee salary respectively. (ii) COUNTR() – To count the number of records present in the CSV file named ‘record.csv’.	
11	# Write a Python function that finds and displays all the words longer than 5 characters from a text file "Words.txt"	
12	WAP to find how many 'f' and 's' present in a text file	
13	Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.	
14	Write a Python function that displays all the words containing @cmail from a text file "Emails.txt".	
15	A list contains following record of a customer: [Customer_name, Phone_number, City] Write the following user defined functions to perform given operations on the stack named status: (i) Push_element() - To Push an object containing name and Phone number of customers who live in Goa to the stack	

	(ii) Pop_element() - To Pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.	
16.	Write a function in Python, Push(SItem) where , SItem is a dictionary containing the details of stationary items– {Sname:price}. The function should push the names of those items in the stack who have price greater than 75. Also display the count of elements pushed into the stack. For example: If the dictionary contains the following data: Ditem={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25} The stack should contain Notebook Pen The output should be: 16 The count of elements in the stack is 2	
17	Consider the tables GARMENT and FABRIC, Write SQL commands for the statements (i) to (iv)	
18	Write SQL commands for (a) to (f) on the basis of Teacher relation	
19	Write a MySQL-Python connectivity code display ename, empno,designation, sal of those employees whose salary is more than 3000 from the table emp . Name of the database is “Emgt”	
20	A table, named STATIONERY, in ITEMDB database, has the following structure: Field Type itemNo int(11) itemName varchar(15) price float qty int(11) Write the following Python function to perform the specified operation: AddAndDisplay(): To input details of an item and store it in the table STATIONERY. The function should then retrieve and display all records from the STATIONERY table where the Price is greater than 120. Assume the following for Python-Database connectivity: Host: localhost, User: root, Password: Pencil	

Q1

Write a function INDEX_LIST(L), where L is the list of elements passed as argument to the function. The function returns another list named indexList that stores the indices of all Non-Zero Elements of L.

```
#sol


def INDEX_LIST(L):
    indexList=[]
    for i in range(len(L)):
        if L[i]!=0:
            indexList.append(i)
    return indexList


L= [12,4,0,11,0,56]
print(INDEX_LIST(L))


'''
OUTPUT
[0, 1, 3, 5]
'''
```

Q2

Write a code in python for a function void Convert (T, N) , which repositions all the elements of array by shifting each of them to next position and shifting last element to first position.

e.g. if the content of array is

0	1	2	3
10	14	11	21

The changed array content will be:

0	1	2	3
21	10	14	11

sol:

```
def Convert ( T,  N):
    for i in range(N):
        t=T[N-1]
        T[N-1]=T[i]
        T[i]=t
    print("LIst after conversion", T)


d=[10,14,11,21]
print("original list",d)
r=len(d)
Convert(d,r)
‘’
OUTPUT:
original list [10, 14, 11, 21]
LIst after conversion [21, 10, 14, 11]
‘’
```

Q3

Create a function showEmployee() in such a way that it should accept employee name, and it’s salary and display both, and if the salary is missing in function call it should show it as 9000

```
'''
def showEmployee(name,salary=9000):
    print("employee name",name)
    print("salary of employee",salary)

n=input("enter employee name")
#s=eval(input("enter employee's salary"))
#showEmployee(n,s)
showEmployee(n)

'''
OUTPUT
enter employee namejohn miller
enter employee's salary6700
employee name john miller
salary of employee 6700
enter employee namesamantha
employee name samantha
salary of employee 9000

'''
```

Q4

Write a program using function which accept two integers as an argument and return its sum.Call this function and print the results in main( )

```
def fun(a,b):
    return a+b

a=int(input("enter no1: "))
b=int(input("enter no2: "))
print("sum of 2 nos is",fun(a,b))


'''
OUTPUT
enter no1: 34
enter no2: 43
sum of 2 nos is 77

'''
```

Q5

Write a definition for function Itemadd () to insert record into the binary file ITEMS.DAT,

(items.dat- id,gift,cost). info should stored in the form of list.

```
import pickle

def itemadd ():

    f=open("items.dat","wb")

    n=int(input("enter how many records"))

    for i in range(n):

        r=int(input('enter id'))

        n=input("enter gift name")

        p=float(input("enter cost"))

        v=[r,n,p]

        pickle.dump(v,f)

        print("record added")

    f.close()


itemadd() #function calling

'''

output

enter how many records2

enter id1

enter gift namepencil

enter cost45

record added

enter id2

enter gift namepen

enter cost120

record added '''
```

Q6

Surya is a manager working in a recruitment agency. He needs to manage the records of various candidates. For this, he wants the following information of each candidate to be stored: -

Candidate_ID – integer –

Candidate_Name – string –

Designation – string –

Experience – float

You, as a programmer of the company, have been assigned to do this job for Surya.

(I) Write a function to input the data of a candidate and append it in a binary file.

(II) Write a function to update the data of candidates whose experience is more than 10 years and change their designation to "Senior Manager".

Write a function to read the data from the binary file and display the data of all those candidates who are not "Senior Manager".

```
#Sol:
import pickle
def input_candidates():
    file=open('candidates.dat', 'wb')
    n = int(input("Enter the number of candidates you want to add: "))
    for i in range(n):
        candidate_id = int(input("Enter Candidate ID: "))
        candidate_name = input("Enter Candidate Name: ")
        designation = input("Enter Designation: ")
        experience = input("Enter Experience (in years): ")
        pickle.dump([candidate_id, candidate_name, designation,experience], file)


def display_non_senior_managers():
    try:
        file=open('candidates.dat', 'rb')
        while True:
            try:
                candidate = pickle.load(file)
                if candidate[2]!= 'Senior Manager':
                    print(candidate[0])
                    print(candidate[1])
                    print(candidate[2])
                    print(candidate[3])
            except:
                break # End of file reached
    except:
            print("No candidate data found. Please add candidates first.")

input_candidates()
display_non_senior_managers()

'''
OUTPUT
Enter the number of candidates you want to add: 3
Enter Candidate ID: 11
Enter Candidate Name: Smith
Enter Designation: Clerk
Enter Experience (in years): 5
Enter Candidate ID: 1001
Enter Candidate Name: Ford
Enter Designation: Analyst
Enter Experience (in years): 12
Enter Candidate ID: 1002
Enter Candidate Name: David
Enter Designation: Senior Manager
Enter Experience (in years): 23
11
Smith
Clerk
5
1001
Ford
Analyst
12
'''
```

Q7

Write a definition for function COSTLY() to read each record of a binary file ITEMS.DAT, find and display those items, which are priced less than 50. (items.dat- id,gift,cost).Assume that info is stored in the form of list

```
#sol
import pickle
def COSTLY():
    f=open("items.dat","rb")
    while True:
        try:
            g=pickle.load(f)
            if(g[2]>50):
                print(g)
        except:
            break
    f.close()



COSTLY() #function calling
'''
output
[2, 'pen', 120.0]
'''
```

Q8

A csv file counties.csv contains data in the following order:

country,capital,code

sample of counties.csv is given below:

india,newdelhi,ii

us,washington,uu

malaysia,ualaumpur,mm

france,paris,ff

write a python function to read the file counties.csv and display the names of all those countries whose no of characters in the capital are more than 6.

```
import csv
def writecsv():
    f=open("counties.csv","w")
    r=csv.writer(f,lineterminator='\n')
    r.writerow(['country','capital','code'])
    r.writerow(['india','newdelhi','ii'])
    r.writerow(['us','washington','uu'])
    r.writerow(['malysia','kualaumpur','mm'])
    r.writerow(['france','paris','ff'])
def searchcsv():
    f=open("counties.csv","r")
    r=csv.reader(f)
    f=0
    for i in r:
        if (len(i[1])>6):
            print(i[0],i[1])
            f+=1
    if(f==0):
        print("record not found")


writecsv()
searchcsv()
'''
output
india
us
malaysia
'''
```

Q9

Write a Program in Python that defines and calls the following user defined functions:

a) add() – To accept and add data of an employee to a CSV file ‘furdata.csv’. Each record consists of a list with field elements as fid, fname and fprice to store furniture id, furniture name and furniture price respectively.

b) search()- To display the records of the furniture whose price is more than 10000.

```
#solution--------------------------------------------
import csv
def add():
    fout=open("furdata.csv","a",newline='\n')
    wr=csv.writer(fout)
    fid=int(input("Enter Furniture Id :: "))
    fname=input("Enter Furniture name :: ")
    fprice=int(input("Enter price :: "))
    FD=[fid,fname,fprice]
    wr.writerow(FD)
    fout.close()
def search():
    fin=open("furdata.csv","r",newline='\n')
    data=csv.reader(fin)
    found=False
    print("The Details are")
    for i in data:
        if int(i[2])>10000:
            found=True
            print(i[0],i[1],i[2])
    if found==False:
        print("Record not found")
    fin.close()

add()
add()
add()
print("Now displaying")
search()

'''
output
Enter Furniture Id :: 10
Enter Furniture name :: chair
Enter price :: 3000
Enter Furniture Id :: 12
Enter Furniture name :: dining table
Enter price :: 20000
Enter Furniture Id :: 13
Enter Furniture name :: study table
Enter price :: 13000
Now displaying
The Details are
12 dining table 20000
13 study table 13000
'''
```

Q10

What is the advantage of using a csv file for permanent storage? Write a Program in Python that defines and calls the following user defined functions:

(i) ADD() – To accept and add data of an employee to a CSV file ‘record.csv’. Each record consists of a list with field elements as empid, name and mobile to store employee id, employee name and employee salary respectively.

(ii) COUNTR() – To count the number of records present in the CSV file named ‘record.csv’.

```
#solution--------------------------------------------
import csv
def ADD():
    fout=open("record.csv","a",newline="\n")
    wr=csv.writer(fout)
    empid=int(input("Enter Employee id :: "))
    name=input("Enter name :: ")
    mobile=int(input("Enter mobile number :: "))
    lst=[empid,name,mobile]
    wr.writerow(lst)
    fout.close()
def COUNTR():
    fin=open("record.csv","r",newline="\n")
    data=csv.reader(fin)
    d=list(data)
    print(len(d))
    fin.close()
ADD()
COUNTR()
'''output
Enter Employee id :: 1001
Enter name :: Anand
Enter mobile number :: 987987981 '''
```

Q11

```
# Write a Python function that finds and displays all the words longer than 5 characters from a text file "Words.txt"

#Sol

def add():

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

    f.write("A paraphrasing tool is an AI-powered solution designed to ")

    f.write("help you quickly reword text by replacing certain words “)

    f.write(“with synonyms or restructuring sentences.")

    f.close()


def disp():

    with open("Words.txt", 'r') as file:

        data=file.read()

        print(data)

def display_long_words():

    with open("Words.txt", 'r') as file:

        data=file.read()

        words=data.split()

        for word in words:

            if len(word)>5:

                print(word,end=' ')



add()

print('file content is')

disp()

print('output should be')

display_long_words()
```

'''

output

file content is

A paraphrasing tool is an AI-powered solution designed to help you quickly reword text by replacing certain words with synonyms or restructuring sentences.

output should be

paraphrasing AI-powered solution designed quickly reword replacing certain synonyms restructuring sentences.

'''

Q12

```
#WAP to find how many 'f' and 's' present in a text file
f=open(r"C:\Users\user\Desktop\cs\networking\firewall.txt")
t=f.read()
c=0
d=0
for i in t:
    if(i=='f'):
        c=c+1
    elif(i=='s'):
        d=d+1
print(c,d)


'''
output
18 41
'''
```

Q13

Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.

```
f=open(r"C:\Users\user\Desktop\cs\networking\firewall.txt")
f1=open("lower.txt","a")
f2=open("upper.txt","a")
f3=open("others.txt","a")
r=f.read()
for i in r:
    if(i>='a' and i<='z'):
        f1.write(i)
    elif(i>='A' and i<='Z'):
        f2.write(i)
    else:
        f3.write(i)

f.close()
f1.close()
f2.close()
f3.close()
```

Q14

```
# Write a Python function that displays all the words containing @cmail from a text file "Emails.txt".

#sol

def add():

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

    f.write("there are many words which contains @email means electronic id of ")

    f.write("various people but questions is to find @cmail accunt which specifies @cmail means corporate mail id")

    f.write("and check occurences of this word")

    f.close()


def disp():

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

    data=f.read()

    print(data)

def show():

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

    data=f.read()

    words=data.split()

    for word in words:

        if '@cmail' in word:

            print(word,end=' ')

    f.close()


add()

print('file content is')

disp()

print('output should be')
```

```
show()


'''
output
file content is
there are many words which contains @email means electronic id of various people but questions is to find @cmail accunt which specifies @cmail means corporate mail idand check occurences of this word
output should be
@cmail @cmail


'''
```

Q15 A list contains following record of a customer:

[Customer_name, Phone_number, City]

Write the following user defined functions to perform given operations on the stack named status:

(i) Push_element() - To Push an object containing name and Phone number of customers who live in Goa to the stack

(ii) Pop_element() - To Pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.

#Ans.

```
status=[]

def Push_element(cust):

    for i in cust:

        if i[2]=="Goa":

            L1=i[0],i[1]

            status.append(L1)

def Pop_element ():


    while status:

        dele=status.pop()

        print(dele)

    else:

        print("Stack Empty")




L=[["Gurdas", "99999999999","Goa"],["Julee", "8888888888","Mumbai"],

["Murugan","77777777777","Cochin"],["Ashmit", "1010101010","Goa"]]

Push_element(L)

print('Stack contains elements are \n ' , status)

print("\n output should be")

Pop_element()
```

```
'''
Output
Stack contains elements are
  [('Gurdas', '99999999999'), ('Ashmit', '1010101010')]

 output should be
('Ashmit', '1010101010')
('Gurdas', '99999999999')
Stack Empty
'''
```

16. Write a function in Python, Push(SItem) where , SItem is a dictionary containing the details of stationary items– {Sname:price}.

The function should push the names of those items in the stack who have price greater than 75. Also display the count of elements pushed into the stack.

For example: If the dictionary contains the following data:
Ditem={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25}

The stack should contain Notebook Pen The output should be: 16 The count of elements in the stack is 2

```
#Sol

stackItem=[]

def Push(SItem):

    count=0

    for k in SItem:

        if (SItem[k]>=75):

            stackItem.append(k)

            count=count+1

    print("The count of elements in the stack is : ",count)




Ditem={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25}

Push(Ditem)


'''

output

The count of elements in the stack is :  2

'''
```

17.Consider the following table GARMENT and FABRIC, Write SQL commands for the statements (i) to (iv)

TABLE GARMENT

GCODE	**DESCRIPTION**	**PRICE**	**FCODE**	**READYDATE**
10023	PENCIL SKIRT	1150	F 03	19-DEC-08
10001	FORMAL SHIRT	1250	F 01	12-JAN-08
10012	INFORMAL SHIRT	1550	F 02	06-JUN-08
10024	BABY TOP	750	F 03	07-APR-07
10090	TULIP SKIRT	850	F 02	31-MAR-07
10019	EVENING GOWN	850	F 03	06-JUN-08
10009	INFORMAL PANT	1500	F 02	20-OCT-08
10007	FORMAL PANT	1350	F 01	09-MAR-08
10020	FROCK	850	F 04	09-SEP-07
10089	SLACKS	750	F 03	20-OCT-08

TABLE FABRIC

FCODE	**TYPE**
F 04	POLYSTER
F 02	COTTON
F 03	SILK
F01	TERELENE

(i) To display GCODE and DESCRIPTION of each GARMENT in descending order of GCODE.

(ii) To display the details of all the GARMENT, which have READYDATE in between 08-DEC-07 and

16-JUN-08 (inclusive if both the dates).

(iii) To display the average PRICE of all the GARMENT, which are made up of fabric with FCODE as

F03.

(iv) To display fabric wise highest and lowest price of GARMENT from GARMENT table. (Display

FCODE of each GARMENT along with highest and lowest Price).

```
Ans .  (i)  SELECT GCODE, DESCRIPTION
            FROM GARMENT ORDER BY GCODE DESC;
       (ii) SELECT * FROM GARMENT
            WHERE READY DATE BETWEEN ’08-DEC-07’
            AND ’16-JUN-08’;
       (iii) SELECT AVG (PRICE)
              FROM GARMENT WHERE FCODE = ‘F03’;
       (iv)  SELECT FCODE, MAX (PRICE), MIN (PRICE)
               FROM GARMENT GROUP BY FCODE;
```

Q18. Write SQL commands for (a) to (f) on the basis of Teacher relation given below:

relation Teacher

No.	**Name**	**Age**	**Department**	**Date of join**	**Salary**	**Sex**
1.	Jugal	34	Computer	10/01/97	12000	M
2.	Sharmila	31	History	24/03/98	20000	F
3.	Sandeep	32	Maths	12/12/96	30000	M
4.	Sangeeta	35	History	01/07/99	40000	F
5.	Rakesh	42	Maths	05/09/97	25000	M
6.	Shyam	50	History	27/06/98	30000	M
7.	Shiv Om	44	Computer	25/02/97	21000	M
8.	Shalakha	33	Maths	31/07/97	20000	F

(a) To show all information about the teacher of history department

(b) To list the names of female teacher who are in Hindi department

(c) To list names of all teachers with their date of joining in ascending order.

(d) To display student's Name, Fee, Age for male teacher only

(e) To count the number of teachers with Age>23.

(f) To inset a new row in the TEACHER table with the following data:

9, "Raja", 26, "Computer", {13/05/95}, 2300, "M"

Ans . (a)

```
SELECT * FROM Teacher

WHERE Department = "History";
```

(b)

```
SELECT Name FROM Teacher

WHERE Department = "Hindi" and Sex = "F";
```

(c)

```
SELECT Name, Dateofjoin

FROM Teacher

ORDER BY Dateofjoin;
```

(d) (The given query is wrong as no. information about students and fee etc. is available.

The query should actually be

To display teacher's Name, Salary, Age for male teacher only)

```
SELECT Name, Salary, Age FROM Teacher

WHERE Age > 23 AND Sex = 'M';
```

(e)

```
SELECT COUNT (*) FROM Teacher

WHERE Age > 23;
```

(f)

```
INSERT INTO Teacher
VALUES (9, “Raja”, 26, “Computer”, {13/05/95}, 2300, “M”);
```

Q19.

Write a MySQL-Python connectivity code display ename, empno,designation, sal of those employees whose salary is more than 3000 from the table emp . Name of the database is “Emgt”

```
import mysql.connector as m

db=m.connect(host="localhost",user="root",passwd="1234",database="Emgt")

c=db.cursor()

c.execute("select * from emp where sal>3000")

r=c.fetchall()

for i in r:

    print(i)
```

Q20.

A table, named STATIONERY, in ITEMDB database, has the following structure:

Field	Type
itemNo	int(11)
itemName	varchar(15)
price	float
qty	int(11)

Write the following Python function to perform the specified operation:

AddAndDisplay(): To input details of an item and store it in the table STATIONERY. The function should then retrieve and display all records from the STATIONERY table where the Price is greater than 120.

Assume the following for Python-Database connectivity: Host: localhost, User: root, Password: Pencil

#Sol

```
def AddAndDisplay():
    import mysql.connector
    mydb=mysql.connector.connect(host="localhost",user="root",passwd="Pencil",
       database="ITEMDB")
    mycur=mydb.cursor()
    no=int(input("Enter Item Number: "))
    nm=input("Enter Item Name: ")
    pr=float(input("Enter price: "))
    qty=int(input("Enter qty: "))
    query="INSERT INTO stationery VALUES ({},'{}',{},{})".format(no,nm,pr,qty)
    mycur.execute(query)
    mydb.commit()
    mycur.execute("select * from stationery where price>120")
    for rec in mycur:
        print(rec)
```

```
AddAndDisplay()

'''
output
Enter Item Number: 10
Enter Item Name: pencil
Enter price: 45
Enter qty: 12
('10001', 'exam board',  800,  10)
('10002', 'parker pen',  750, 5)
'''
```