

SISHYA SCHOOL, HOSUR

LAB MANUAL

COMPUTER SCIENCE (083)

PYTHON PROGRAMS

1. Develop a python program to read a text file line by line and display each word separated by #.

```
*exp1.py - D:/SISHYA/PRACTICAL/XII/exp1.py (3.8.6)*
File Edit Format Run Options Window Help
#open file in read mode
f=open("sample.txt","r")
#Read the content of file
content=f.readlines()
for line in content:
    words=line.split()
    for word in words:
        print(word+"#",end=" ")
f.close()

Ln: 13 Col: 0
```

OUTPUT

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15 :37:30) [MSC v.1927 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: D:/SISHYA/PRACTICAL/XII/exp1.py =====
This# is# sample# program# reads# line# by# line#
```

2. Develop a python program to create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.

PROGRAM

```
#Write data into binary file
import pickle
student=[]
f=open("student.dat", 'wb')
ans='y'
while ans.lower()=='y':
    roll=int(input("Enter Roll Number:"))
    name=input("Enter the Student Name:")
    student.append([roll,name])
    ans=input("Add more (Y)?")
    pickle.dump(student,f)
f.close()

#Read data from binary file
f=open("student.dat", 'rb')
student=[]
while True:
    try:
        student=pickle.load(f)
    except EOFError:
        break
ans='y'

#Search record
while ans.lower()=='y':
    found=False
    s=int(input("Enter Roll No to search:"))
    for s in student:
        if s[0]==roll:
            print("Name : ",s[1])
            found=True
            break
    if not found:
        print("Record not found...")
    ans=input("Search more?(Y) :")
f.close()
```

OUTPUT

```
Enter Roll Number:1
Enter the Student Name:Ramesj
Add more (Y) ?Y
Enter Roll Number:2
Enter the Student Name:Mahesh
Add more (Y) ?N
Enter Roll No to search:2
Name : Mahesh
Search more? (Y) :?N
```

3. Develop a python program to implement a PUSH () and POP () operations of a stack using a list data structure.

PROGRAM

```
stack = []
# Push operation
def push(stack, item):
    stack.append(item)
    print(f"{item} pushed to the stack.")

# Pop operation
def pop(stack):
    if len(stack) == 0: # Directly check if the stack is empty
        print("Stack is empty! Cannot pop.")
        return None
    else:
        item = stack.pop()
        print(f"Popped: {item}")
        return item

# Display the stack
def display(stack):
    if len(stack) == 0: # Directly check if the stack is empty
        print("Stack is empty!")
    else:
        print("Current stack:", stack)
```

```
# Main program
def main():
    while True:
        print("\nChoose an operation:")
        print("1. Push")
        print("2. Pop")
        print("3. Display stack")
        print("4. Exit")
        choice = input("Enter your choice (1-4): ")

        if choice == "1":
            item = input("Enter the item to push: ")
            push(stack, item)
        elif choice == "2":
            pop(stack)
        elif choice == "3":
            display(stack)
        elif choice == "4":
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

main()
```

OUTPUT

```
Choose an operation:  
1. Push  
2. Pop  
3. Display stack  
4. Exit  
Enter your choice (1-4): 1  
Enter the item to push: 25  
25 pushed to the stack.
```

```
Choose an operation:  
1. Push  
2. Pop  
3. Display stack  
4. Exit  
Enter your choice (1-4): 1  
Enter the item to push: 46  
46 pushed to the stack.
```

```
Choose an operation:  
1. Push  
2. Pop  
3. Display stack  
4. Exit  
Enter your choice (1-4): 3  
Current stack: ['25', '46']
```

```
Choose an operation:
```

- 1. Push
- 2. Pop
- 3. Display stack
- 4. Exit

```
Enter your choice (1-4) : 3
```

```
Current stack: ['25', '46']
```

```
Choose an operation:
```

- 1. Push
- 2. Pop
- 3. Display stack
- 4. Exit

```
Enter your choice (1-4) : 2
```

```
Popped: 46
```

```
Choose an operation:
```

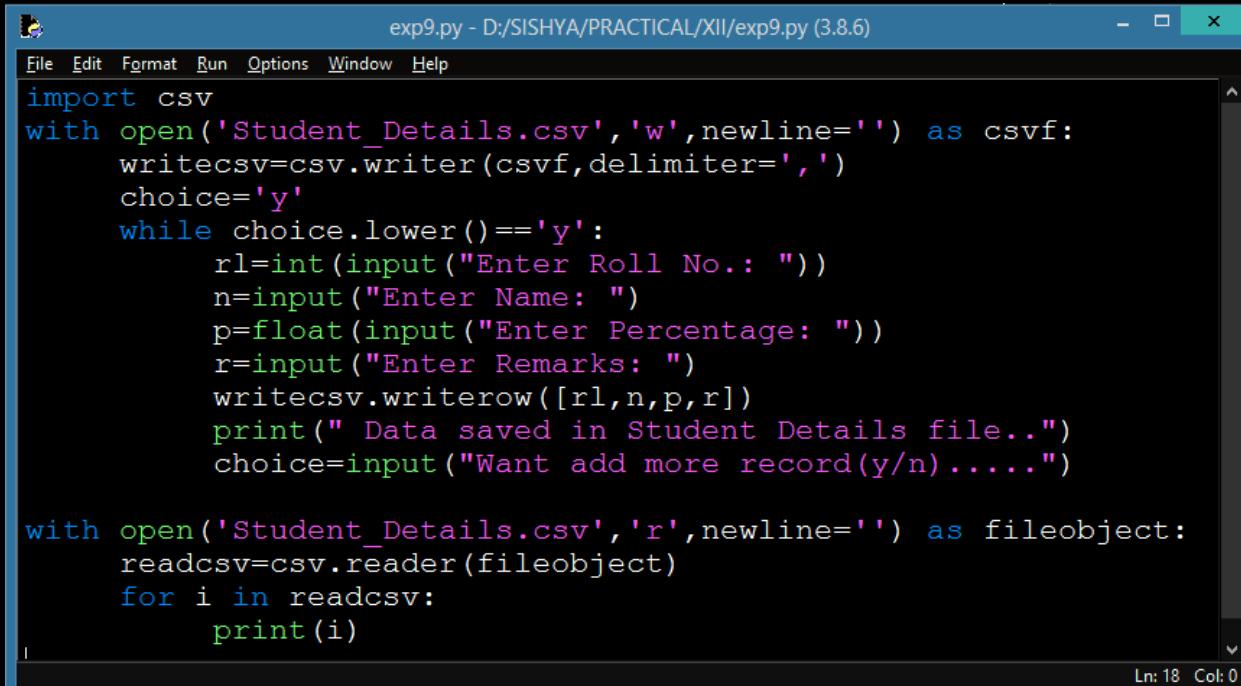
- 1. Push
- 2. Pop
- 3. Display stack
- 4. Exit

```
Enter your choice (1-4) : 4
```

```
Exiting program. Goodbye!
```

4. Develop a python program to perform read and write operation onto a “student.csv” file having fields as roll number, name stream and percentage.

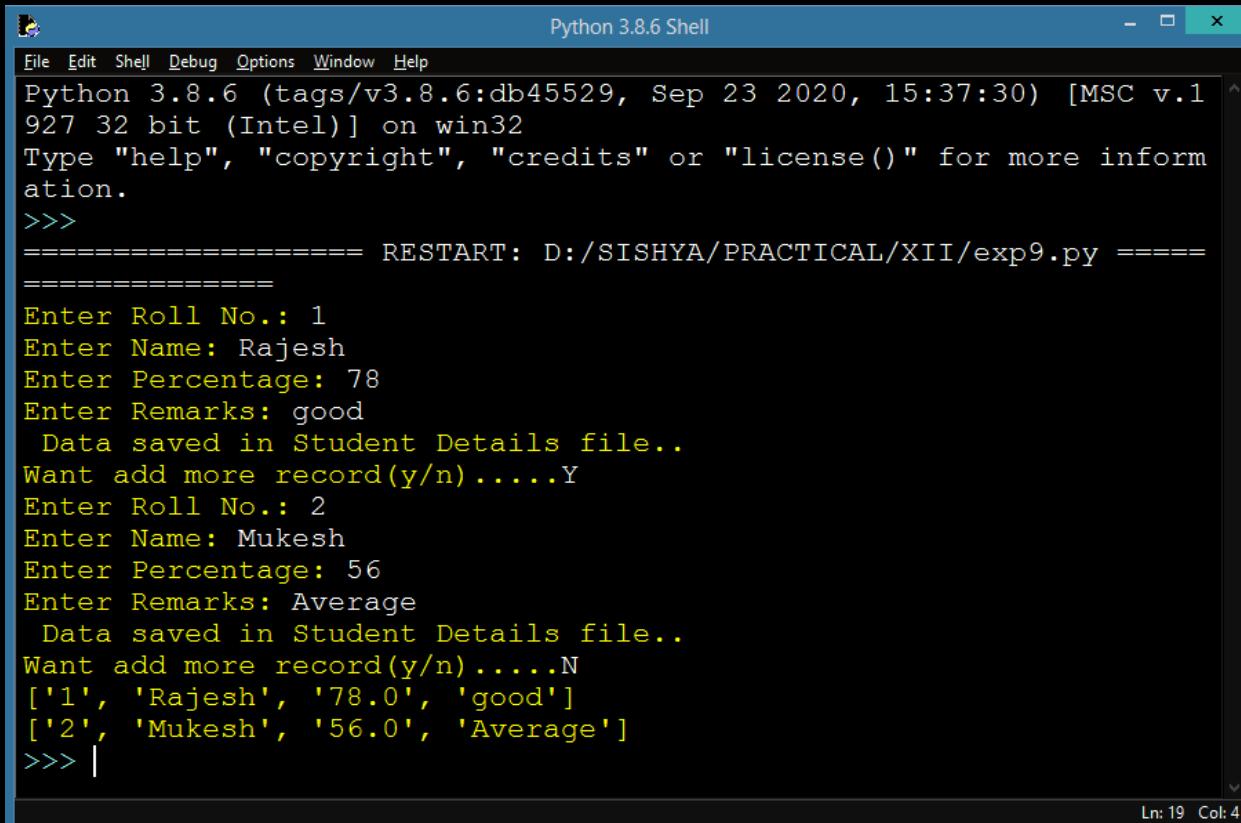
PROGRAM



```
exp9.py - D:/SISHYA/PRACTICAL/XII/exp9.py (3.8.6)
File Edit Format Run Options Window Help
import csv
with open('Student_Details.csv','w',newline='') as csvf:
    writecsv=csv.writer(csvf,delimiter=',')
    choice='y'
    while choice.lower()=='y':
        rl=int(input("Enter Roll No.: "))
        n=input("Enter Name: ")
        p=float(input("Enter Percentage: "))
        r=input("Enter Remarks: ")
        writecsv.writerow([rl,n,p,r])
        print(" Data saved in Student Details file..")
        choice=input("Want add more record(y/n).....")

with open('Student_Details.csv','r',newline='') as fileobject:
    readcsv=csv.reader(fileobject)
    for i in readcsv:
        print(i)
Ln: 18 Col: 0
```

OUTPUT



```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:37:30) [MSC v.1
927 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more inform
ation.
>>>
===== RESTART: D:/SISHYA/PRACTICAL/XII/exp9.py =====
=====
Enter Roll No.: 1
Enter Name: Rajesh
Enter Percentage: 78
Enter Remarks: good
Data saved in Student Details file..
Want add more record(y/n).....Y
Enter Roll No.: 2
Enter Name: Mukesh
Enter Percentage: 56
Enter Remarks: Average
Data saved in Student Details file..
Want add more record(y/n).....N
['1', 'Rajesh', '78.0', 'good']
['2', 'Mukesh', '56.0', 'Average']
>>> |
Ln: 19 Col: 4
```

MYSQL PROGRAMS

1. Write the SQL commands for (i) to (iii) and write the output of the (iv) on the basis of table Employee.

Relation : Employee

ENO.	Name	Age	Department	Dateofadm	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

i	To create a table employee.
ii	To insert the given 5 records.
iii	To display the names of all employees whose department is neither 'computer' nor 'history'.
iv	Give the output of the following SQL statement. Select COUNT(distinct Department) from Employee;

- I. CREATE TABLE EMPLOYEE (ENO INT,NAME VARCHAR(25),AGE INT,DEPARTMENT VARCHAR(25), DATEOFADM DATE, SALARY INT,SEX CHAR(1));
- II. INSERT INTO EMPLOYEE VALUES
(1,"JUGAL",34,"COMPUTER","10/01/97",12000,"M");
- III. SELECT NAME FROM EMPLOYEE WHERE DEPARTMENT NOT IN
("COMPUTER","HISTORY");
- IV.

COUNT(DISTINCT(DEPATMENT))
3

2.

Write the SQL commands for (i) to (iii) and write the output of the (iv) on the basis of table FRESH.

TABLE NAME: FRESH

ITEMCODE	ITEMNAME	SCODE	QTY
1001	POTATO	11	50
1002	ONION	11	50
1003	TOMATO	12	25
1004	CAULIFLOWER	13	50
1005	GARLIC	14	20
1006	GINGER	14	25

i	To create a table FRESH
ii	To insert the above records into the table.
iii	To display all the values of the table where the itemname has at least one character “o”.
iv	Give the output of the following SQL statement. SELECT ITEMNAME FROM FRESH WHERE SCODE>11 AND QTY<20;

- I. **CREATE TABLE FRESH (ITEMCODE INT,ITEMNAME VARCHAR(30),SCODE INT, QTY INT);**
- II. **INSERT INTO FRESH VALUES (1001,"POTATO",11,50);**
- III. **SELECT * FROM FRESH WHERE ITEMNAME LIKE "%O%";**
- IV.

ITEMNAME
NONE

3.

Write the SQL commands for (i) to (iv) on the basis of table STUDENT

TABLE: STUDENT

ADMNO	NAME	CLASS	SEC	STREAMID
1101	NAVEEN	XII	A	10
1102	CHETAN	XII	A	10
1103	IMRAN	XI	C	30
1104	SHAILENDRA	XII	B	20
1105	TEJAS	XI	A	10
1106	ZOYA	XI	B	20
1107	SWATI	XII	D	40

i	To create a table student
ii	To insert all the records shown in the table.
iii	To display the names of all students who belongs to class XII and their STREAMID is greater than 10.
iv	To sort the given table in ascending order of the student name.

- i. **CREATE TABLE STUDENT (ADMNO INT, NAME VARCHAR(25),CLASS VARCHAR(20),SEC CHAR (1), STREAM ID INT);**
- ii. **INSERT INTO STUDENT VALUES (1101,"NAVEEN","XII","A",10);**
- iii. **SELECT NAME FROM STUDENT WHERE CLASS="XII" AND STREAMID>10;**
- iv. **SELECT * FROM STUDENT ORDERBY NAME;**

4.

Write the SQL commands for (i) to (iv) on the basis of table ACCESSORIES

TABLE: ACCESSORIES			
No	Name	Price	Id
A01	MOTHER BOARD	12000	S001
A02	HARD DISK	5000	S002
A03	MOUSE	450	S001
A04	KEYBOARD	700	S001
A05	LCD	10000	S003
A06	LCD	11000	S002
A07	HARD DISK	5500	S002
T08	MOUSE	400	S003
T09	KEYBOARD	650	S001
T10	MOTHER BOARD	13000	S003

i	To create a table accessories.
ii	To insert the given records into the table
iii	To display the name and price of all the accessories whose price is less than 4000.
iv	To display the price of the product whose id is either “S001” or “S003”.

- i. **CREATE TABLE ACCESSORIES (NO VARCHAR(30),NAME VARCHAR(30),PRICE INT,ID VARCHAR(30));**
- ii. **INSERT INTO ACCESSORIES VALUES (“A01”,“MOTHER BOARD”,12000,“S001”);**
- iii. **SELECT NAME,PRICE FROM ACCESSORIES WHERE PRICE <4000;**
- iv. **SELECT PRICE FROM ACCESSORIES WHERE ID=“S001” OR ID = “S003”;**