

## LAB CYCLE - 7

### Experiment No :1

Date :18/12/2024

### Aim :

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

### Pseudocode :

Main:

OPEN "1.txt" in read mode ('r') as f  
Initialize an empty list l

READ the first 4 lines from the file and append each to the list l  
CLOSE the file f

Initialize an empty list l2  
FOR each word (wd) in list l:  
    STRIP any leading/trailing whitespace from wd and append it to list l2

PRINT the list l2

### Method :

Functions	Description	Syntax
open()	Opens a file, returns a file object. Used for reading or writing to files.	open(filename, mode)
close()	Closes an opened file. After closing, you can't perform further operations on the file.	file.close()

### **Source Code :**

expt1.py

```
f=open("1.txt",mode='r')
l=[]
l.append(f.readline())
l.append(f.readline())
l.append(f.readline())
l.append(f.readline())
f.close()
l2=[]
for wd in l:
    l2.append(wd.strip())
print(l2)
```

1.txt

```
hello
olleh
hi
ih
```

### **Output :**

```
['hello', 'olleh', 'hi', 'ih']
```

**Result :** The program is successfully executed and the output is verified.

## **Experiment No :2**

**Date:** 18/12/2024

**Aim :**

Python program to copy odd lines of one file to other.

**Pseudocode :**

Main:

    OPEN "1.txt" in read mode ('r') as f

    READ all lines from the file and store them in list l

    CLOSE the file f

    Initialize an empty list l2

    FOR each word (wd) in list l:

        STRIP any leading/trailing whitespace from wd and append it to list l2

    OPEN "2.txt" in write mode ('w') as f2

    Initialize an empty list l3

    FOR i in range from 0 to length of l2, with a step of 2:

        Append l2[i] to list l3

    FOR each word (wd) in list l3:

        WRITE wd to file f2 followed by a newline character

    CLOSE the file f2

    OPEN "2.txt" in read mode ('r') as f2

    READ all lines from the file and store them in list l4

    CLOSE the file f2

    PRINT the contents of list l4

**Source Code :**

1.txt

hello

olleh

hi

ih

expt2.py

```
f=open("1.txt",mode='r')
l=f.readlines()
f.close()
l2=[]
for wd in l:
    l2.append(wd.strip())

f2=open("2.txt",mode='w')
l3=[]
for i in range(0,len(l2),2):
    l3.append(l2[i])
for wd in l3:
    f2.write(wd+"\n")
f2.close()

f2=open("2.txt",mode='r')
l4=f2.readlines()
f2.close()
print(l4)
```

**Output :**

['hello', 'hi']

**Result :** The program is successfully executed and the output is verified.

## Experiment No :3

Date: 18/12/2024

### Aim :

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

### Pseudocode :

```
FUNCTION read_csv_as_strings(file_name)
    TRY
        OPEN file with file_name in read mode
        INITIALIZE csv_reader for the file
        FOR each row in csv_reader
            PRINT row
    EXCEPT FileNotFoundError
        PRINT "The file 'file_name' was not found."
    EXCEPT Exception as e
        PRINT "An error occurred: e"

IF __name__ IS "__main__"
    SET file_name TO "thirdqs.csv"
    CALL read_csv_as_strings(file_name)
```

### Source Code :

```
import csv

def read_csv_as_strings(file_name):
    try:
        with open(file_name, 'r') as file:
            csv_reader = csv.reader(file)
            for row in csv_reader:
                print(row)
    except FileNotFoundError:
        print(f'The file '{file_name}' was not found.')
    except Exception as e:
        print(f'An error occurred: {e}')

if __name__ == "__main__":
    file_name = "thirdqs.csv"
    read_csv_as_strings(file_name)
```

**Output :**

```
['Student ID', 'Name', 'Course', 'semester']  
['2286', 'Anand', 'MCA', '3']  
['2264', 'Ananthan', 'MCA', '2']  
['2292', Hari, 'MCA', '1']
```

**Result :** The program is successfully executed and the output is verified.

## Experiment No :4

Date: 18/12/2024

### Aim :

Write a Python program to read specific columns of a given CSV file and print the content of the columns.

### Pseudocode :

```
FUNCTION read_specific_columns(file_name, column_indices)
    TRY
        OPEN file with file_name in read mode
        INITIALIZE csv_reader for the file
        FOR each row in csv_reader
            CREATE selected_columns as a list containing values from row at indices in
column_indices
            PRINT selected_columns
    EXCEPT FileNotFoundError
        PRINT "The file 'file_name' was not found."
    EXCEPT IndexError
        PRINT "One of the column indices is out of range."
    EXCEPT Exception as e
        PRINT "An error occurred: e"

IF __name__ IS "__main__"
    SET file_name TO "thirdqs.csv"
    SET column_indices TO [0, 2]
    CALL read_specific_columns(file_name, column_indices)
```

### Source Code :

```
import csv

def read_specific_columns(file_name, column_indices):
    try:
        with open(file_name, 'r') as file:
            csv_reader = csv.reader(file)
            for row in csv_reader:
                selected_columns = [row[index] for index in column_indices]
                print(selected_columns)
    except FileNotFoundError:
        print(f"The file '{file_name}' was not found.")
    except IndexError:
        print(f"One of the column indices is out of range.")
```

```
except Exception as e:
    print(f'An error occurred: {e}')

if __name__ == "__main__":
    file_name = "thirdqs.csv"
    column_indices = [0, 2]
    read_specific_columns(file_name, column_indices)
```

### **Output :**

```
['Name', 'Country']
['Alice', 'USA']
['Bob', 'UK']
['Charlie', 'Canada']
```

**Result :** The program is successfully executed and the output is verified.



## Experiment No :5

Date: 18/12/2024

### Aim :

Write a Python program to write a Python dictionary to a csv file. After writing the CSV file, read the CSV file and display the content.

### Pseudocode :

PROMPT user for the number of entries to add and store in num\_entries  
INITIALIZE empty lists: sid, name, course, sem

```
FOR i from 0 to num_entries - 1
    PROMPT user for student id and store in stdid
    PROMPT user for student name and store in names
    PROMPT user for course name and store in crse
    PROMPT user for semester and store in sems
    APPEND stdid to sid
    APPEND names to name
    APPEND crse to course
    APPEND sems to sem
```

CREATE dictionary data with keys 'Student ID', 'Name', 'Course', 'Semester' and corresponding lists

```
OPEN 'thirdqs.csv' in write mode
    CREATE csv.DictWriter with fieldnames from data.keys()
    WRITE header to the CSV file
    FOR each index in the range of the length of data['Name']
        CREATE row by mapping keys to data values at the current index
        WRITE row to the CSV file
```

```
OPEN 'thirdqs.csv' in read mode
    CREATE csv.DictReader for the file
    PRINT "CSV file contents:"
    FOR each row in the CSV reader
        PRINT row
```

## Source Code :

```
import csv
num_entries = int(input("Enter the number of entries you want to add: "))
sid = []
name = []
course = []
sem = []
for i in range(num_entries):
    stid = input(f"Enter student id: ")
    names = input(f"Enter the student name: ")
    crse = input(f"Enter the name of the course: ")
    sems = input("Enter the semester: ")
    sid.append(stid)
    name.append(names)
    course.append(crse)
    sem.append(sems)
data = {
    'Student ID': sid,
    'Name': name,
    'Course': course,
    'Semester': sem
}
with open('thirdqs.csv', mode='w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames=data.keys())
    writer.writeheader()
    for i in range(len(data['Name'])):
        row = {key: data[key][i] for key in data}
        writer.writerow(row)
with open('thirdqs.csv', mode='r') as file:
    reader = csv.DictReader(file)
    print("\nCSV file contents:")
    for row in reader:
        print(row)
```

## Output :

```
Enter the number of entries you want to add: 2
Enter student id: 12345
Enter the student name: John Doe
Enter the name of the course: Computer Science
Enter the semester: Fall 2024
Enter student id: 67890
Enter the student name: Jane Smith
Enter the name of the course: Mathematics
Enter the semester: Spring 2024
```

CSV file contents:

```
{'Student ID': '12345', 'Name': 'John Doe', 'Course': 'Computer Science', 'Semester': 'Fall 2024'}
```

```
{'Student ID': '67890', 'Name': 'Jane Smith', 'Course': 'Mathematics', 'Semester': 'Spring 2024'}
```

**Result :** The program is successfully executed and the output is verified.