

Task 6. Implement various text file operation.

Aim:-

To write a python program implement various text file operations.

6.1:-

Algorithm:

1. Write to a file:

- Define writefile (filename) function:
 - open a file named "log.txt" in write mode,
 - write the following text to the file.
 - Error objects are thrown when runtime errors occur. The error object can also be used as a base object for user-defined exceptions.
 - close the file.

2. Read from a file:

- Define readfile (filename) function:
 - open the file specified by filename in read mode using a with statement.
 - Read the entire content of the file.
 - Print the content.

3. Execute the Program:

- call writefile ("write") to write the predefined text to "log.txt".
- call readfile ("text") to attempt to read from a file named "text" and print its content.

Program :-

```
def writefile (filename):  
    f=open ("log.txt","w")  
    f.write ("Error objects are thrown when runtime error  
-so occur. The error object can also be used as a base  
object for user-defined exceptions")  
    f.close()
```

```
def readfile (filename):
```

Output:-

Error objects are thrown when runtime errors occur. The error object can also be used as above object for user defined exceptions

```
with open(filename, "r") as file:  
    content = file.read()  
    print(content)  
write(file, "write")  
read(file, "text")
```

6.2:-

You have a text file log.txt containing logs of a system. write a function that counts the number of lines containing the "ERROR".

Algorithm:

1. Initialize Error Counter:

- Define the function count_error_lines(filename):
 - Initialize error_count to 0.

2. Open and Read file:

- Open the file specified by filename in read mode using a with statement.

3. Check each line for "ERROR":

- Loop through each line in the file:
 - If the line contains the word "ERROR", increment error_count by 1.

4. Return Error Count:

- After reading all the lines, return the value of error_count.

5. Execute the program:

- Call count_error_lines("log.txt") to count the number of lines with the word "ERROR" in the file "log.txt".
- Print the result with the message: "Number of lines with 'ERROR': {error_lines}".

Program:-

```
def count_error_lines(filename):  
    error_count = 0  
    with open(filename, "r") as file:  
        for line in file:  
            if "ERROR" in line:
```

Output:-

Number of lines with 'ERROR' is 2

```
error_count += 1  
return error_count  
error_lines = count_error_lines("log.txt")  
print(F"Number of lines with 'ERROR': {error_lines}")
```

log.txt

"Error objects are thrown when runtime errors occur."

The Error object can also be used as a base object for user-defined exceptions."

6.3:- You need to write a report containing the details (name, departments) of the employee in list. Write a python function that writes this report to a file named employee-report.txt.

Algorithm:

1. Create employee Data:

- Define the function write_employee_report(filename)
- Create a list employees containing dictionaries, each with "name" and "department" keys for individual employees.

2. Open file for writing

- Open the file specified by filename in write mode using a with statement.

3. Write employee data to file:

- Loop through each employee in the employee's list:
 - For each employee, format a string as "Name: {employee['name']} , department: {employee['department']}
 - Write the formatted string to the file, followed by a newline character (\n).

4. Execute the program:

- Call write_employee_report("employee-report.txt") to write the employee data to the file "Employee-report.txt".

Program:-

```
def write_employee_report(filename):
    employees = [
        {"name": "Alice", "department": "HR"},
        {"name": "Bob", "department": "Engineering"},
        {"name": "Charlie", "department": "Finance"}]
```

]

with open(filename, "w") as file:

```
    for employee in employees:
```

```
        line = f"Name: {employee['name']}, department:  
        {employee['department']} \n"
```

```
        file.write(line)
```

Example usage,

```
write_employee_report("employee_report.txt")
```

VELTECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	8/12/11

Result:- Thus, the python program Implement various text file operations was successfully executed and the output was verified

output:-

Name : Alice, department : HR

Name : Bob, department : Engineering

Name : Charlie, department : Finance