

Date 8/10/2028

Task - 6.1

Implement various test file operations.

AIM :- To write a python program implement various
Text file operations.

ALGORITHM :-

1. Write to a file :

- Define `write_file(filename)` function:
- Open a file named "log.txt" in write mode.
- Write the following text to the file
- Close the file.

2. Read from a file :

- Define `read_file(filename)` functions :
- Read the entire content of the file
- Print the content.

3. Execute the program :

- Call `write_file("write")` to write pre-defined text to "log.txt".
- Call `read_file("text")` to attempt to read from a file named "text" and print its content.

PROGRAM :-

```
def write_file(filename):  
    F = open('log.txt', 'w')  
    F.write('Error objects are thrown when runtime  
    errors occurs. The Error object can  
    also be used as a base object for  
    user-defined exceptions!')  
    F.close()
```

```
def read_file(filename):  
    with open(filename, 'r') as file:
```

Output

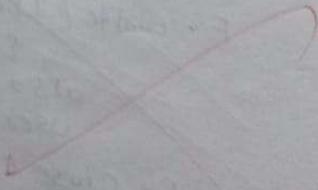
Error objects are thrown when runtime error occurs.

object at 0x1000000000000000
method from file /lib/libc.so.6
new in /lib/libc.so.6
old in /lib/libc.so.6
old at 0x1000000000000000

object at 0x1000000000000000
method from file /lib/libc.so.6
new in /lib/libc.so.6
old in /lib/libc.so.6

object at 0x1000000000000000
method from file /lib/libc.so.6
new in /lib/libc.so.6
old in /lib/libc.so.6

(more and more objects are listed, get
rid of them by pressing q)



Content = file.read()

Print (Content)

write file ('write')

read file ('text')

10/9/25

Task - 6.2

AIM :- To write a python function that counts the number of lines containing the word 'Error' in a log file.

ALGORITHM :-

1. Initialize Error Counter:
→ Define the function count_Error_lines (filename):
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.
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")
→ Print the result with the message.

PROGRAM :-

```
def count_Error_lines(filename):  
    Error_Count = 0  
    with open(filename, "r") as file:  
        for line in file:  
            if 'ERROR' in line:  
                Error_Count += 1  
    return Error_Count  
Error_lines = count_Error_lines("log.txt")  
Print("Number of lines with 'ERROR':", Error_lines)
```

OUTPUT :-

16000 - 800 = 15200

(15200) 1000

(15200) 200 200

Number of lines with error : 2

Date : 10/9/25

Topic 6.3

AIM :- To write a python function that generates an employee report with names and departments and save it 'employee-report.txt'.

ALGORITHM :-

1. Create Employee Data.
→ Define the function `write_employee_report(filename)`
2. Open file for writing:
→ Open the file specified by `filename` in write mode using a with statement.
3. Write an employee data to file:
→ Loop through each employee in the Employee list.
4. Execute the program.

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']},\n"
            line += f"Department : {employee['department']} \n"
            file.write(line)
```

OUTPUT :-

Name : Alice, Department : HR.

Name : BOB, Department : Engineering

Name : Charlie, Department : Finance

file.write(lire)

write - Employee-report ("Employee-report.txt")

VEL TECH - CSE	
EX NO.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	15
DATE WITH DATE	

RESULT :- Thus, the Python program implement various text file operations was successfully executed and the output was verified.