10/9/25

Task 6:- implement various text file operation.

Aim:- To write a Python program implement various text file operations is executed & verified Successfully.

Problem 6.1:-

You need to write the sentence "Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions" into a text file named log.txt. implement a Function that Performs this task.

Algorithm:-

1. write to a File:-
   - Define writefile (file name) function:
   - open a file named "log.txt" in write mode.
   - write the Following text to File:-
   - "Error objects are thrown when runtime errors occur.
   - close the file.

2. Read From a File:-
   - Define readfile (file name) 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

Output :- Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions.

text to "log.txt".

- Call readfile ("text") to attempt to read from a file named "text" and Print its content.

Program 6.1 :-

```
def writefile (file name):
    f = open ("log.txt","w")
    f.write ("Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions")
    f.close ()
def readfile (file name):
    with open (file name, "r") as file:
    content = file.read ()
    Print (content)
write file ("write")
read file ("text")
```

Problem 6.2 :-

You have a text file log.txt containing logs of a system. write a function that counts the no. of lines containing the word "ERROR!"

Algorithm :-

1. initialize Error Counter :-

- Define the function count_error_lines (file name):
- 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. Execute the Program:-

- Call Coont_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 6.3:-

```
def Coont_error_lines (file name):
    error_Coont = 0
    with open (file name, "r") as file:
        For line in file:
            if "ERROR" in line:
                error_Count += 1.

    return error_Coont
error_lines = Coort_error_lines ("log.txt")
Print f ("No. of lines with 'ERROR : {error_
                                    lines}")
```

log txt

"Error objects are thrown when runtime Error occur.

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

Algorithm:-

1. Create Employee Data:

- Define the function write-employee-report (filename):
- Create a list employees containing dictonaries, 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 employees 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 File "employee_reporter.txt".

Program 6.3:-

```
def write_employee_report (filename):
    employee = [
    {"name": "Alice", "department": "HR"},
    {"name": "Bob", "department" : "Engineering"},
```

output :-

Name : Alice, Department : HR

Name : Bob, Department : Engineering

Name : Charlie, Department : Finance

```
{"name": Charlie", "department": "Finance"}
]
with open (file name, "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")
```

VEL TECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	4
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20	
SIGN WITH DATE	

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