Task: 6 :- Implement various text file operation.

Aim:-

To write a python program Implement various text file operations.

6.1:- You need to write the sentence "Error objects are thrown when return runtime errors occur. The Error object can also be used as a a base object for user-defined expceptions" into a text file named log.txt. Implement a function that performs this task.

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 occurs - The Error object can also be used as a base object for used-difined 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 statment.

     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".

Output:

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

- 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
             errors occurs. The error objects can also
             be used as a base objects for user-
             defined exceptions")

    f:close()
def readfile(filename):
    with open(filename,"r") as file:
    content=file.read()
     print(content)
writefile("write")
readfile("text")
```

6.2:- You have text file log.txt containing logs of a system. Write a function that counts the number of lines containing the word "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}"

Output:

Number of lines with 'ERROR' is 2

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_line = count_error_lines("log.txt")
print(f"Number of Lines with 'ERROR': {error_lines}")
```

log.txt

"ERROR object are thrown when runtime Error occurs. The Error object can also be used ~~be~~ as a base object for user-difined exceptions."

6.3:- You need to write a report containing the detials (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 fuction write_employee_report (filename): Create a list employee containing dictionary; each with "name" and "department" key for individual employees.

2. Open file for writing:
   - open the file specified by filename in ~~mode~~ write mode using a with statement.

3. Write Employee Data to file:
   - Loop through each employee in the employee 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".

Output:

```
Name: Alice , Department: HR
Name: Bob, Department: Engineering.
Name: Charlie, Department: Finace.
```

Program:

```
def write_employee_report(filename):
  employee= [
   {"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")
```

VEL TECH	
EX No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (3)	
VIVA VOCE (3)	
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	

Result:-

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