Task 6: Implement various text file operation

Aim: To write a python program implement various text file operations

problem 6.1:

You need to wire 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:

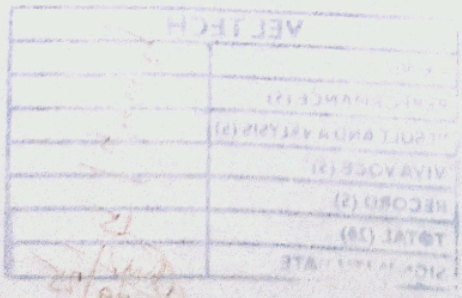
1. write to a file:
   - Define write file(filename) function:
     - Open a file named "log.txt" in write mode.
     - write the following text to the file:
       - "Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions.
     - Close the file.
2. Read from a file:
   - Define read file (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 write file("write") to write the predefined text to "log.txt"
   - Call read file ("text") to attempt to read from a file named "text" and print is content.

Program 6.1:

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

Output:

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



Task 6:

Aim: To

Operations

problem 6.1:

You need

runtime e

object for

log.txt. En

Algorithm:

1. Write

- Define
  - Op
  - W

2. Read f
   - De

3. Execute th
   - Ca
   - Call
     "t

Program 6.1:

```
def writ
    f =
    f.
    Er
    in
    f.
def readfi
```

write file ("write")
read file ("text")

problem 6.2:

You have a text file log.txt containing logs of 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 (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. 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}".

6.2)

```
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(f"Number 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."

Output:

Number of lines with 'ERROR' is 2

Output:

Name: Alice, Department: HR

Name: Bob, Department: Engineering

Name: Charlie, Department: Finance

Problem 6.3:

You need to write a report containing the details (name, department) of the employee in list. write a python function that writes this report to a file named employee_report.text.

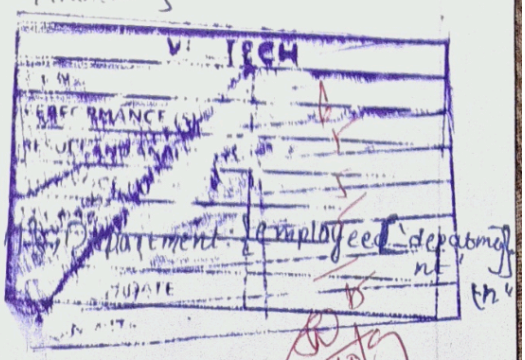
Algorithm:

1. Create Employee Data:
   - Define the follow 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 list:
     - For each employee, format a strings 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 6.3:

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

# Example usage:
write_employee_report("employee_report.txt")
```



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