

Task 6:- Implement various text file operations

Aim:-

To write a python program implement various text file operation.

G.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" in a text file named log.txt. Implement a function that performs this task.

Algorithm:-

1. write to a file:

- Define writefile(filename) function:
- open 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 text "log.txt"
- Call readfile ("text") to attempt to read from a file named "text" and print its 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.

Exception

• An exception is an object that is thrown when something goes wrong. It is a type of error object.

• When something goes wrong, an exception is thrown. This leads to a loop of code execution and exception handling.

• A red circle is drawn around the word "exception".

Program 6.1

```

def writefile(filename):
    f = open("log.txt", "w")
    f.write("error objects are thrown when runtime errors occur")
    f.close()

def readfile(filename):
    with open(filename, "r") as file:
        content = file.read()
        print(content)

writefile("write")
readfile("fext")

```

6.2

you have a textfile 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 - takes (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.

output

Number of lines with "error" is 0.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

70.

71.

72.

73.

74.

75.

76.

77.

78.

79.

80.

81.

82.

83.

84.

85.

86.

87.

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

4. Return - Error Count:

- After reading all the lines, returns 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's: {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:
```

```
                error-count += 1
```

```
    return error-count
```

```
error-lines = count-error-lines ("log.txt")
```

```
print("Number of lines with 'error': {error-lines}")
```

log.txt

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

Problem 6.3:- You need to write a report containing the details
(name, departments) of 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 `employees` list:
 - For each employee, format a string as `"name": {"Employee": ["name"], "Department": ["Employee": ["name"]] }"`.
 - 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 employment 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"}]
```

Output:-

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.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	9
VIVA VOCE (5)	+
RECORD (5)	
TOTAL (20)	16
SIGN WITH DATE	16 / 20

OK

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